Abstract

This Thesis applies complex spatial filters to the front end filtering to a computer vision framework for object recognition and scene categorization. This involves careful filter design in the Fourier domain based on discrete frame properties. Biological plausibility of the suggested filtering is compared against a common model found in the computer vision literature. The designed complex filter bank is equipped with focus-of-attention operators. Specifically, two possible keypoint detection methodologies are examined and compared with state of the art keypoint detection methods. This includes an investigation of scale-estimation methods. In addition, three image patch descriptor arrangements are proposed to sample the complex filter responses, and an initial evaluation of categorization performance is undertaken. Next, the spatial pooling arrangement of the best performing descriptor is further optimised and the performance of different complex filter bandwidths is examined in class separation tasks. A further study is conducted on the effects of a Winner-Take-All (WTA) approach to modifying filter responses before pooling. A thorough evaluation of descriptor performance is undertaken to reveal any advantages or disadvantages from a variety of perspectives. Next, the clustering behaviour of descriptors of various types is inspected in the descriptor feature space. A reverse look-up of visual words attempts to relate clustering behaviour to descriptor performance. Typical grouping approaches, such as spatial pyramids, are then compared with a novel method for coupling visual words in which a linear kernel SVM learns class separability. A final evaluation on this stage is presented and discussed, leading to conclusive arguments about the importance of careful approaches to word-pairing for good-quality categorization.
7.3.1 Experiments ........................................... 102
7.4 Pairing Up Visual Words ................................. 106
7.5 Coupling Visual Words ................................. 107
  7.5.1 A Dictionary of Paired Visual Words ............... 108
  7.5.2 Decision Functions for Coupling .................. 110
7.6 Experiments ........................................... 113
  7.6.1 Evaluation of Coupling Functions ................. 114
  7.6.2 Pascal VOC 2011 .................................. 114
  7.6.3 Caltech101 ........................................ 115
  7.6.4 Pascal VOC 2007 .................................. 116
  7.6.5 Pairs of LogFoveal Descriptors .................. 117

8 Conclusions and Future Work ............................ 118
  8.1 Multilevel Complex Filtering ......................... 118
     8.1.1 Future Work .................................... 119
  8.2 Keypoint Detection .................................. 120
     8.2.1 Future Work .................................... 121
  8.3 Descriptors ........................................... 121
     8.3.1 Future Work .................................... 122
  8.4 Visual Words ........................................ 123
     8.4.1 Future Work .................................... 124
  8.5 Spatial Information .................................. 124
     8.5.1 Future Work .................................... 125
List of Figures

1.1 This image illustrates which the cortical regions are thought responsible for object recognition tasks and where are located in the human brain which. The diagram on the right shows the processing stages and timings for each cortical area. Reproduced from [32] ................................. 2
1.2 A representation of the cells’ structure in the retina. Reproduced from [28] ................................. 3
1.3 A representation of two types of ganglion cells. On the right; the inhibitory area of the cell is the blue surrounding area with the central red part generating excitatory responses. On the left, the attributes of the cell have been reversed in order to demonstrate different contrast preference. The two input signals have two spatially oriented frequencies and two contrast types. It is noticeable that both cells have no orientation selectivity, but respond to contrast differently. In addition, both cells are tuned to respond to similar spatial frequencies [28] ................................. 4
1.4 The organization in the LGN of six cell layers. Reproduced from [28] ................................. 5
1.5 Representation of simple cells receptive fields in the primary visual cortex V1. The coloured bars on the right indicate orientation preference of these cells. In the receptive field each colour indicates a region where simple cells are selective to an orientation as indicated by the colour bars. Reproduced from [75] ................................. 6
1.6 The top part of this figure illustrates one hypothesis that explains how LGN receptive fields are organized to form the input of an orientation selective simple cell. At the top, three ganglion cells are organized vertically to form the input of a simple cell which can be represented by a 2D Gabor function. On the bottom a representation of a complex cell is provided. Here, the complex cell does not change its response by position shift in the receptive field, but merely responds to orientation [55]. This is illustrated by introducing 4 phase offsets to Gabor (simple cells) and then computing the magnitude of the Gabor filters. The magnitude response, in this case may also referred to as oriented energy of the complex cell [11].

1.7 There are 4 suggested models of border ownership. (A) Feed forward mechanisms (B) Lateral propagation in cortex, (C) Feedback from a higher cortical area and (D) Integration fields (or grouping cell model) where ellipses symbolize receptive fields and the square is stimulus. Reproduced from [103].

1.8 This is an illustration of concave and convex shapes, introduced as stimuli to V4 area. A comparison with a tuned model with the same shape selectivity is done in order to “parameterize” an approximate computational neuronal model [19].

1.9 This Figure illustrates the classification modules that form a forward processing pipeline. Each Chapter in this Thesis is related to one of the modules.

2.1 The DoG scale space is obtained by blurring an image repeatedly with Gaussians and computing the difference between these blurred representations. The image is resized to make the spanning of several octaves efficient. By halving the size of the image, a drop of 1 octave is achieved in the spatial frequency domain. Reproduced from [63].
2.2 A keypoint is detected in scale space as previously described. This point is remapped to the original image to calculate the image gradients. The gradients are computed by derivative operators and weighted by an isotropic Gaussian window represented by the blue circle above. This region is split into 4 quadrants where each is sampled using a histogram of oriented gradients (HOG). Each bin of the HOG represents a gradient orientation, and there are 8 bins in total. Having acquired 4 HOGs as illustrated on left side of figure, histograms are generated which are incorporated into the SIFT patch descriptor. The figure illustrates only 4 sub-patches instead of 16 which are the implementation. Reproduced from [63].

2.3 HMAX model utilizes a series of layers of simple and complex cells to perform object recognition. Starting from the bottom, each layer accumulates information from the previous layer using weighted sum and max pooling as indicated. The top layer approximates a response of V4 and inferotemporal cortical area (IT) [82].

3.1 This figure illustrates an unoptimised frame. The purple curves represent the log-normal distributions in Fourier space. The blue line corresponds to the unoptimised frame.

3.2 The red curves show the optimized spacing where no oscillations appear along the curves. The final sigma values are: $\sigma_\theta = 0.346$, $\sigma_\rho = 0.612$ used to the final configuration of the wavelet dictionary.

3.3 A Fourier representation of the optimized filter set. All scales are independently mapped to collect the spatial responses. Above, for illustration purposes, each filter is normalised to have a peak amplitude of 1, and the filters are combined by a max() projection onto the $u_y, u_x$ Fourier plane.
3.4 A typical one dimensional signal “unravels” phase properties of a 1-dimensional signal from a 2D phase-space to the spatial domain. The horizontal axis on the top two plots show pixel locations and the \( y \) axis shows pixel intensity and phase angle output respectively.

3.5 A test image that reveals the behaviour of the cosine and sine parts of phase. This is equivalent to the original convolution of the signal with a complex filter pair.

4.1 This Figure illustrates the Gaussian scale space of a simple one-dimensional signal. A line from the middle of the height of the image which contains rectangular white boxes has been extracted. The blue line highlights the extracted one dimensional signal. The output of increasing size filter produces the scale space in this example. It is clearly shown that local maxima appear along the scale in this space which often are described as blobs. For a 2D signal (image) a 3D space is formed where the responses around the maxima in this space form spherical regions also known as blobs.

4.2 This test image is part of the scale optimization process where consecutive 2D Gaussian functions of increasing \( \sigma \) values with range \((1, ..., 32)\) and added levels of noise \((0.001, ..., 0.3)\) evaluate the estimated scale of the examined implementations.

4.3 On top, the original image taken from the Pascal VOC 2011 class aeroplane. The left column shows a series of the \( T_{ss} \) response at the 2nd, 3rd, and 4th scales. The right column shows different scales of \( V_{ss} \).

4.4 This graph illustrates the deviation of the ground truth (red dashed diagonal line), from the estimated scales using \( T_{ss} \) (sum) and \( V_{ss} \) (product) to reconstruct scales from the steered log normal responses.

4.5 This graph presents the produced error of the estimated scale relative to the ground truth. Also the learned curve is highlighted, showing only small deviation from the best possible correction.
4.6 The diagonal red lines show the optimal scale target in log$_2$ space. The $y$ axis shows the estimated scale using 10 intervals of random Gaussian noise. The error bars indicate the average standard deviation for the [0.1%, 1%, 10%, 30%] noise levels.

4.7 Test Sequences:(a) Textured scene with zoom and rotation changes, (b) Increasing blur, (c) Structured grey-scaled scene with zoomed and rotated depictions, (d) Viewpoint change, (e) Decreasing light conditions, (f) Increased JPEG compression.

4.8 The matching scores refer to the percentage of the key-points that fall into the accepted regions by the homographies. This figure presents the performance of the key-points on the Bark sequence, which is a series of images, rotated and zoomed, of the texture of a tree’s bark.

4.9 The Bike sequence contains six images depicting road bikes under different blurring conditions. The images are gradually blurred starting from the first which the finer blur and ending to the sixth which is the coarsest.

4.10 This figure presents the matching score of the indicated detector by the legend (top right). These images depict a boat rotated and rescaled with the global scale and rotation change are conveyed by the homographies.

4.11 The Car sequence contains six images depicting cars under different illumination conditions. The images are gradually dimmed starting from the first which the brightest and ending to the sixth, which is the darkest.

4.12 A scene picturing graffiti under a variety of out-of-plane tilts. This sequence also incorporates scale changes in order to fit on the frame the geometrically transformed scene.

4.13 The House scene is a series of compressed images in which each contains an increased compression factor relatively to the previous one. All images are processed with the same compression protocol to produce “jpeg” files.
5.1 Descriptor Construction: A circular sampling area is defined by a keypoint’s scale and divided into 8 sectors. Each sector is sampled to produce a histogram of complex (cell’s) orientations (magnitude of complex Gabor filter outputs). The phase of the Gabor outputs can also be embedded in the descriptor as information which describes local structure [43]. The bottom left figure shows how max pooling is performed across orientations in a Gabor jet. On the bottom right, the grouping of position shifts and spatial structure is shown.

5.2 Descriptor illustration: These two versions of descriptors are biologically inspired from (V1) to approximate responses of complex cells (64D) and 4 variants of simple cells (256D) [55],[75].

5.3 As shown, 17 spatial pooling kernels with positional shifts are used to define spatial regions for generating descriptor elements by pooling directional filter outputs. The fixation point lies on the center of the union of all spatial kernels. Vertical is y Cartesian axis and horizontally the x axis where these have been converted to polar coordinates to map the pooling kernels. For illustration purposes, all 17 kernels have been overlaid into a single image to illustrate the foveal arrangement and the spatial relationships at each lobe against the others.

5.4 LogFoveal descriptor (D5) optimization on Pascal VOC 2011. The dashed blue circle indicates an area of asymptotic behaviour where no global maximum can be achieved.

5.5 An illustration of maximal suppression across orientations. The coloured circles indicate patches of different orientations. A sample of eight orientations at a point \((x, y)\) is processed to eliminate all non-maximum orientation responses. This process is performed independently for each scale, rather than across scales.

5.6 (a) An example of good class separation in which the correspondent AUC metric is above the diagonal dashed line. (b) illustrates the opposite effect where the examined class has higher distances distribution yielding a low AUC metric.
5.7 A graphic representation of the band optimization sampling the responses of Complex log-normal filters on a regular grid descriptor (D7). The colour-bar values indicate the AUC value range displayed on the image.

5.8 This figure illustrates the performance of foveal pooling using two weighting function (Gauss, log-normal). The AUC metrics are sorted in ascending order over categories to present a readable graphic of detailed descriptor performance.

6.1 An example of annotated images in VOC Database along with their bounding boxes per object class.

6.2 This is the distance distribution of the SIFT descriptor with no angular shifting of the descriptor. The broad coloured area along the curves indicates the deviation of distance per cluster centre (visual word).

6.3 A dictionary of Complex log-normal wavelets with optimised bandwidth produces its scale space which, in turn, is applied in the classification framework. The descriptors are grouped into 500 clusters to produce the result shown in this figure.

6.4 The SIFT descriptor includes rotation invariance. The same number of clusters (500) organise the descriptors into groups.

6.5 This figure illustrates the cluster separation using Log-Foveal descriptors on the produced log-normal nonlinear scale space.

6.6 The cluster distances severely change when maximal suppression is introduced to the Log-Foveal descriptors.

6.7 A representation using two principal components of the cluster distributions on a 2D projection. It is visible that the cluster scattering does not obey to a uniform distribution.

6.8 This plot shows the cluster scattering of the SIFT-like (rectangular grid (D7)) descriptors sampled on a log-normal scale space with optimised bands to yield the best class separability.
6.9 SIFT descriptors are rotation invariant in this example where the cluster of these descriptors are projected above. It is noticeable how the orientation invariance has changed the cluster distribution.

6.10 In this example, Log-Foveal descriptors sampled by the raw outputs of the log-normal responses are further clustered. Such descriptor clusters are illustrated above.

6.11 The maximal suppression is introduced in the log-normal outputs which are sampled by the Log-Foveal descriptors. The cluster distribution is much different from the previous examples where here the cluster centres are co-organised into elongated lobes.

6.12 Visual Words reverse engineered from complex log-normal magnitude to original patches. The patches have been acquired from the Pascal VOC 2011 database.

6.13 Visual Words reverse engineered from (SIFT’s) Gaussian scale space with two directional derivatives to original patches. The patches have been acquired from the Pascal VOC 2011 database.

7.1 This is a graphic representation of how the image is subdivided into regions. Three levels of a spatial pyramid are presented with increasing resolution of each level of the pyramid.
7.2 This is a geometric interpretation of how the SVM classifier separates two classes in a binary problem. The coloured dots represent instances of two classes in the feature space. The red ones correspond to negative labeled classes and the blue ones to positive labeled classes. The separation margin between the classes is indicated by an orange line along with its width $\frac{2}{||w||}$. The borders of the margin are described by the illustrated equation which a compact form is presented by the Equation (7.3). The purple dashed lines correspond to $\xi$, introduced to handle classification noise along the borders. The coloured dots which are crossed by the black lines are called support vector as they are the instances which support the margin. Altered from [26] . . . . . . . . . . . . . . . . . . . . . . . . . . . . 100

7.3 This figure presents the classification accuracy of the Gauss-Foveal (D3 & D4) descriptors on Caltech 101. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 103

7.4 This is graphical representation of the classification performance of the LogFoveal (D5 & D6) descriptors on Caltech 101. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 104

7.5 Starting from left to right: (a) illustrates how a kernel may be constructed using conditional histograms. (b) This figure shows the concept of first and second maxima. Each colour represents a different pair, where the size of the coloured blobs illustrates the 1st and 2nd maxima. (c) Once a candidate category is selected, we focus on the maximum votes that belong to this category. Finally, the ratios of these pairs are selected from highest to lowest. . 109

7.6 This figure shows two word co-occurrences before pairing. The size of the circles indicates the scale at those locations, and the colour represents a unique word ID. The pair detection is performed by flagging the word IDs of a single pair. At this point, several word co-occurrences might happen. The kernel on the right of the figure, as defined in Equation (7.10) arranges the co-occurrences and selects the maximum values along the largest dimension. The remainders are added as probabilistic votes into the histogram bin which indicates the examined pair. . . . . . 111
7.7 (a) Illustrates a word pair comprised of two words $w_A$ and $w_B$. These words convey information, such as the effective descriptor radius ($\sigma_A$ and $\sigma_B$), their relative Euclidean distance ($d_{AB}$), which is used to derive the paired-words. (b) Illustrates the number of times that a single pair occurred (as described in Equation (7.10)) before adding up the votes.

7.8 This is the only experiment in which the $\alpha$ parameter of Equation (7.11) is varied. A red line sets the baseline performance of the sigmoid function where the blue curve shows which values of $\alpha$ increase or reduce the accuracy. This test has been performed on the class “Cow” of Pascal VOC 2011. There is a clear trend for the specific range of $\alpha = (8, 10.5)$ which can yield better performance for this class. The nature of this effect is class specific.

7.9 In (a) a comparison of spatial pyramids (blue) and pair (red) with descriptors produced by keypoint locations. (b) The same comparison is illustrated, but this time a grid sampling approach has been implemented. In both (a) and (b) the “confidence” represents the unit-standard deviation envelope estimated over 10 runs.
**List of Tables**

4.1 Summary table of repeatability per detector on the Oxford Affine Test Data. Scores refer to the percentage of matched keypoints per Test Set after being averaged from the previous figures. Each detector is configured to output approximately 1200 keypoints. .....

4.2 Summary table of repeatability per detector on the Oxford Affine Test Data. The table presents the repeatabilities’ standard deviation over all affine transformations for each test set. Although the standard deviations are low for the Harris Laplacian detector note that there is a large bias shown in Figures 4.8-4.13. .....

5.1 Complex Gabor wavelet performance for descriptors (D1 & D2) on Pascal VOC 2010 Train Set compared with SIFT descriptor. .....

5.2 Band optimization of the log-normal Filters using (SIFT & D7) on the Pascal VOC 2011 Train Set. Band refers to the radial frequency crossing point of the filters as defined in Chapter 3 and illustrated by Figure 3.2. .....

5.3 This table presents the results of log-normal filters after applying the maximal suppression (This is descriptor D8) to the top three best performance bands. Specifically, from the Table 5.2 the top 3 average AUCs are selected and processed again. The maximal suppression treatment boosts the performance, on average. .....

53

54

68

69

71
5.4 Descriptor Performance AUC% of (D3, D4, D5 and D6) using descriptor distances in Pascal VOC. Two pooling function are used where either Gaussian weights (D3 & D4) or log-normal weights (D5 & D6) accumulate gradient responses.

5.5 This table presents the summaries of the different descriptor arrangements tested on Caltech 101 and Pascal VOC 2011. It is notable that the best performing arrangement is the LogFoveal. Filter bank optimization can give further improvements to overall performance.

7.1 Spatial Pyramids and SVM Accuracy% in Pascal VOC 2011.
7.2 Overall accuracies (averaged) across all 4 tests.
7.3 This table clearly shows that the best performance is obtained by $S_1$ ($\alpha = 1$). The average performance of the examined measures was tested on the Train set which contains approximately 5K images.
7.4 Detailed performance evaluation on Pascal VOC 2011.
7.5 Summary of table (7.5). These are the accuracies averaged over all classes, with standard deviation reflecting variability across all classes.
7.6 Average accuracies per method for Caltech 101.
7.7 Detailed performance on Pascal VOC 2007. Overall, the pair approach achieves 66.84% against 62.21% for the spatial pyramids.
7.8 Detailed performance on Pascal VOC 2011. In test the Sparse Log Foveal descriptors (D6) use smaller number of pairs than SIFT.
Project Contributions by the Author

- Design, configuration and coding of a 2D Wavelet transform in Fourier Space.
- Complex filter set optimization methodology.
- Keypoint detection and Scale estimation methodology.
- Design and optimization of a fast spatial pooling descriptor.
- Proposed, implemented and tested a keypoint detector and scale-estimator for a complex log-normal filter bank.
- Developed and applied a methodology for assessing the variability of categorization performance with different training sets in moderate-sized image databases.
- Thorough study of the effects of non-maximal suppression during descriptor construction.
- Thorough investigation of descriptor feature space and interactions of clusters within that space.
- Proposal of categorization opponency to select word pairings to achieve sparse image encoding.
- Published Work In Chapter 5: Ioannis Alexiou, Anil Anthony Bharath, Zeynep Engin. Categorisation Performance Using V1 Spatial cRF Simulations. AVA/BMVA Meeting on Biological and Computer Vision 2011 (poster).
Chapter 1

Introduction to Early Biological Vision

Recently, there has been considerable interest in developing scalable object recognition algorithms capable of searching large databases rapidly [87]. In the field of visual search a lot of effort has been put on the discovery any parallels between machine vision and biological visual systems. The main reason for this interest is two-fold: (i) it is considered very difficult to create algorithms that are scalable, yet simultaneously capable of ranking visual similarity or categories of query with high accuracy, and (ii) there is a realisation that biological vision is vastly more efficient, in terms of learning and recognition performance, than the equivalent computer counterparts.

The next Sections will (a) summarise some basic models of spatial processing in the early visual cortex (b) loosely compare the SIFT algorithm with spatial processing in early biological vision (c) suggest a mapping between the underlying representation of SIFT (an isotropic Gaussian scale-space with local partial derivative estimates) and “complex cells” (d) propose a new method, based on sampling Gabor-jet representations to produce a descriptor that is as scalable as the SIFT method. The purpose this is to determine whether improved performance can be achieved by closely related biologically inspired filters comparing to Gaussian derivatives than SIFT framework uses.

Although dynamical models are currently favoured for accurately modelling cortical neurons in computational neuroscience, it is unquestionably the case that V1 pyramidal neurons exhibit strong direction selec-
tivity, spatial phase invariance and response inhibition; [53], [20], [18] such behaviour can be relatively easily modelled by 2D complex Gabor kernels, nonlinear rectification and strategies such as winner-take-all. It would, thus, be very interesting to compare the behaviour of descriptors based on Gabor functions with accepted methods such as SIFT.

1.1 Visual Pathway in Mammalian Cortex

The mammalian visual system can interpret the complex visual information that is found in the real-world very easily. A single glance is sufficient for the visual cortex to extract valuable characteristics on the location, shape, colour and texture of an object. Also, the adaptability of biological vision to light and terrain conditions is far from achievable, even in state-of-the-art recognition systems\(^1\). Specifically human visual perception performs surprisingly well in poor illumination, and in the presence of highly cluttered scenes. This observation, in addition to the considerations of performance, remains a challenge to the best algorithms, and it is generally considered that visual search to the level of human performance will require approximately many years’ worth of research.

![Figure 1.1: This image illustrates which the cortical regions are thought responsible for object recognition tasks and where are located in the human brain which. The diagram on the right shows the processing stages and timings for each cortical area. Reproduced from [32]](http://www.google.com/mobile/goggles)

\(^1\)http://www.google.com/mobile/goggles
1.2 Retina

The first part of the vision’s central nervous system is comprised of five types of cells: photoreceptors, bipolar cells, ganglion cells, horizontal cells, and amacrine cells [85]. The eye is the first sensory organ that receives and transforms light into neuronal signals. Specifically, the retina is a layer of neurons lying in the eye where light stimulates the photoreceptors which transmit information for further processing. There are two types of photoreceptors in the retina: rods encoding luminance and cones encoding three main photopigments red, green and blue (more accurately wavelengths of light) [99]. The next layer is occupied by bipolar and horizontal cells; amacrine cells moderate, and provide lateral inhibition, in ganglion cells [99], [85]. Finally, dendritic ganglion cells transmit their activation signal to the optic nerve.

![Diagram of the retina showing the structure of photoreceptors, bipolar cells, and ganglion cells.](image)

**Figure 1.2:** A representation of the cells’ structure in the retina. Reproduced from [28].
1.2.1 Modelling Spatial Properties of Retinal Cells

Recent research has sought to create extensive models of the dynamical properties of ganglion cell outputs of biological retinas [78]. Although these models aim to capture the spiking rates of retinal ganglion cells as closely as possible based on experimental data, they are not easily applicable to practical computer vision. More recent work has pointed to a very rich set of spatial and temporal channels being produced by the retina [55].

At the other end of complexity, the most basic models of retinal behaviour aim to capture how the peak firing rates in a ganglion cell relate to the light intensity that is incident on the photoreceptors that are nearby. For example, receptive fields of ganglion cells can be approximated by Laplacian of Gaussian or a Difference of Gaussians (DoG) [60],[17] spatial patterns.

![Figure 1.3: A representation of two types of ganglion cells. On the right; the inhibitory area of the cell is the blue surrounding area with the central red part generating excitatory responses. On the left, the attributes of the cell have been reversed in order to demonstrate different contrast preference. The two input signals have two spatially oriented frequencies and two contrast types. It is noticeable that both cells have no orientation selectivity, but respond to contrast differently. In addition, both cells are tuned to respond to similar spatial frequencies [28].](image)
1.3 Lateral Geniculate Nuclei (LGN)

Retinal ganglion cells send information primarily to the LGN where further processing is applied by the local neuronal arrangements. This area is arranged in 6 layers which are occupied by three different cell types [74]. Magnocellular (M) cells are located in layers 1 and 2 which have large size with respect to the size of the cell body, the dendritic structure, and their receptive field. These cells respond to movement, depth and low contrast conditions. Parvocellular (P) cells are small in size and located in the layers 3, 4, 5 and 6 [9]. Responses from red and green cones are handled by P cells for colour perception and fine spatial details. Also, P cells respond to, and are thought to encode, long and medium wavelength (related to light spectrum not spatial frequency) information based on their dendritic structure. Koniocellular (K) cells are the smaller cell type in LGN and are distributed along the 6 layers and in between of M and P cells [47]. K cells encode short wavelengths as well as the responses from the blue cones.

![Diagram of LGN layers and cell types](image)

**Figure 1.4:** The organization in the LGN of six cell layers. Reproduced from [28].

Because the LGN is a deep brain structure, less is known about the behaviour of cells within it. However, it is thought that at least some of the responses may be similar to the retinal ganglion cells. On the other hand, it is known that V1 (primary visual cortex) sends projections to the LGN as well, and there may be other brain areas that modulate responses based on attention or other factors [98].
1.4 Primary Visual Cortex V1

Retinal responses are completely different from those in the V1 area. Retinal responses are conveyed to this region via the Lateral Geniculate Nuclei. The exact cell types and their complete functional descriptions are still very relevant topics of research. However, spatial receptive fields that display strong response variation with visual stimuli [75] are widely cited [90], and orientation selectivity has even been reproduced in experiments on unanaesthetised animals [21]. The observation dates back to the work of Hubel and Wiesel [53] who proposed that cells in V1 response to stimuli of higher complexity than ganglion cells in retina and LGN. Structurally, although blobs [75] exist in specific regions in V1 which appear to respond to colour and ocular dominance, simple and complex cells in V1 can be excited by bars or line stimuli.

![Figure 1.5: Representation of simple cells receptive fields in the primary visual cortex V1. The coloured bars on the right indicate orientation preference of these cells. In the receptive field each colour indicates a region where simple cells are selective to an orientation as indicated by the colour bars. Reproduced from [75].](image)

Simple cells respond to visual structures with a geometrically linear appearance such as bar or edges. These cells have a strong orientation preference: as the projected bars are rotated in space, then different regions of the receptive field are stimulated. A simple cell has excitatory and inhibitory regions, as retinal ganglion cells do, but respond over a larger visual angle than the latter. D.H. Hubel and T.N. Wiesel have proposed a slightly generalised simple cell model for explaining spatial
receptive fields of oriented simple cells in which lateral geniculate cells in a linear array arrangement form the inputs of these oriented simple cells [52]. The excitatory and inhibitory regions are elongated in comparison with retinal cells. Figure 1.6 illustrates an organization of LGN cells forming the input to a simple cell.

Figure 1.6: The top part of this figure illustrates one hypothesis that explains how LGN receptive fields are organized to form the input of an orientation selective simple cell. At the top, three ganglion cells are organized vertically to form the input of a simple cell which can be represented by a 2D Gabor function. On the bottom a representation of a complex cell is provided. Here, the complex cell does not change its response by position shift in the receptive field, but merely responds to orientation [55]. This is illustrated by introducing 4 phase offsets to Gabor (simple cells) and then computing the magnitude of the Gabor filters. The magnitude response, in this case may also referred to as oriented energy of the complex cell [11].

A simple cell can be also simulated by a linear combination of DoG filters but this approximation is harsh due to spatial arrangement of the DoGs. John G. Daugman has suggested that simple cells can also be approximated by the use of two-dimensional Gabor Filters [29]. A Gabor filter can be configured to the desired orientation and shape (bar or edge) selectivity. Gabor parameters allow one to approximate various simple cells in the receptive field such as orientation and phase. The former enables the steerability of the filter by rotating a kernel which imitates orientation selective cells [31], [82], [53]. The latter (phase) can
simulate the shape selectivity of the cell such as bright or dark bars and edges. Bars and edges are also sometimes referred to as primitive features which are considered essential elements contained in more complex object structures.

Hubel and Wiesel also described cells that were spatially phase invariant. Their experiments indicate that these complex cells are selective to orientation and remain active with small bar shifts in the cortex. They suggested that these could be modelled by a collection of a group of simple cells with the same orientation selectivity, but different spatial phases of response. The organization of the cells in V1 has been described as being loosely columnar where each orientation sensitive column “monitors” a retinal region. There are several layers in V1 processing orientation which communicate with pyramidal cells generating excitatory signals of activated neurons.

1.5 Prestriate Cortex V2

V2 is the second major area in the visual cortex [55]. Generally, V2 is not as thoroughly studied as V1, but recent experiments have revealed specific characteristics. Experiments suggest that receptive fields in this area are stimulated by more complex shape characteristics than V1, although there is also a feedback procedure between V1 and V2. In particular, Nan R. Zhang and Rudiger von der Heydt [103] have discovered a mechanism that encodes the ownership of V1 features to an object. Using repeated experiments involving square stimuli, a significant intracortical function of V2 was uncovered. This mechanism is termed border ownership: cells are tuned to assign edges to a region. This mechanism has been likened to Gestalt laws, employing principles of similarity and convexity, and are generally accepted as principles of visual grouping. Furthermore, in [103] it is suggested that the figure-ground organization effect can be explained by the assignment of borders to objects in V2 which become salient in the cortex through V2 receptive fields.

Jay Hegde and David C. Van Essen have suggested that V2 is selective to slightly more complex features than V1. Experiments [46] have compared the shape selectivity of elementary shapes between V2 and V4. Unlike V1 spatially receptive fields, there is no known simple parametrization of the spatial responses invoking maximum firing rate in
these experiments. It is claimed by [3] that the receptive field properties observed in V2 might perform grouping and curvature estimation [103]. Both theories suggest that early grouping and segmentation functions occur, without proposing a model that approximates the receptive fields of V2.

1.6 Extrastriate Visual Area V4 and Inferotemporal Cortex

Anterior to V2 lies another visual cortical area (V4) where more complex grouping properties have been studied recently. Experiments in that region have shown more complex shape selectivity [46],[19]. It has been proposed that V4 and IT might produce high order informative descriptions or contain specific shape selective receptive fields which significantly enhance object recognition in vision. Although other studies have shown that visual attention modulates the tunings in V4, it is concluded that grouping based on saliency of features is performed [19]. Comparing this
model closely resembles the shape of a critical region in the stimuli that elicit high responses.

Testing population selectivity for boundary conformation Model C2 units can successfully fit the V4 population selectivity data and can generalize to V4 responses outside the training set. For each V4 neuron, we divided the main stimulus set randomly into two nonoverlapping groups (a training and a testing set) in a standard cross-validation procedure (see METHODS). Figure 5 shows correlation coefficient histograms for training and testing over the population of V4 neurons. The median correlation coefficient between the neural data and the C2 unit responses was 0.72 (explained variance 52%) on the training set, and 0.57 (explained variance 32%) on the test set over sixfold cross-validation splits of the dataset. However, because the stimulus set is inevitably correlated, the test set correlation coefficients are inflated. The full distributions of the model parameters can be found in supplemental figure S2.

Much of the variance in V4 neuron responses may be unexplainable due to noise or uncontrolled factors. Pasupathy and Connor (2001) estimated the noise variance by calculating the average expected squared differences across stimulus presentations. The estimated noise variance averaged 41.6% of the total variance. Using this estimate, on the training set the model

![Figure 1.8: This is an illustration of concave and convex shapes, introduced as stimuli to V4 area. A comparison with a tuned model with the same shape selectivity is done in order to “parameterize” an approximate computational neuronal model [19].](image)

theory with the shape selectivity does not contradict both findings, but provides another perspective of grouping in V4.

1.7 Thesis Structure

Although some of this thesis describes biologically inspired computational models, the rest of the thesis is organised primarily around the specific components of an object categorization pipeline. The Chapters are organised as follows.

1. **Front end filtering** multilevel analysis based on filters (related Chapter 3).

2. **Keypoint detection** and scale estimation (related Chapter 4).

3. **Descriptor sampling.** Local patch description designs (related Chapter 5).

4. **Clustering.** Visual word construction and internal evaluation (related Chapter 6).
5. **Spatial information encoding.** Embedding spatial information to visual word encoding (related Chapter 7).

6. **Learning - Classification.** Preferably a choice of a margin classifier (related Chapter 8).

![Diagram](image.png)

**Figure 1.9:** This Figure illustrates the classification modules that form a forward processing pipeline. Each Chapter in this Thesis is related to one of the modules.

In Chapter 2, a brief review of object recognition methods in computer vision, including methods that are used in machine vision (industrial inspection), medical imaging, and methods based on hierarchical learning. More attention is given to methods that are closer to biological vision, such as SIFT and HMAX models and Gabor jets.

In Chapter 3, the first stage processing is an important ingredient for a object recognition framework where this chapter suggests an alternative to widely used gradient fields. A design for 2D complex filters is proposed to enhance the information acquired from gradient fields. An optimisation method is suggested to tune the complex filter responses for scale and orientation estimation. Also, an interpretation of the merits of such filtering is given.

In Chapter 4, scale detection schemes are proposed suitable for a scalable object recognition. Two approaches are suggested and benchmarked.
with widely used keypoint detectors. Specific focus is given on the scale estimate correction for the two suggested approaches. In addition, a simpler method of scale estimation is presented and evaluated using the Oxford Affine Detector dataset [68].

In Chapter 5, pooling arrangements [100] are introduced and evaluated. Specifically, different descriptor pooling arrangements are tested on the using the outputs of Chapter 3.

In Chapter 6, clustering approaches are introduced as a means to reduce the number of descriptors in a database. The widely used $k$-means method is used to produce 500 clusters for the descriptors, which are discussed in Chapter 5, followed by an investigation of the properties which are conveyed by such an approach.

In Chapter 7, spatial information is incorporated as an extension to local patch descriptors. Two methods, spatial pyramids [58] and coupled visual words [105] are discussed. On top of the discussed methods, machine learning approaches such as kernel machines and SVMs are employed to learn class separability using spatial information.

In Chapter 8, conclusions and future work are presented for each of the previous chapters. An overall conclusion to the findings of this thesis is provided.
Chapter 2

Object Recognition in Computer Vision

Until relatively recently, many real-world object recognition methods used in automated inspection were template based, inflexible to different scales and heuristically optimized. Beyond using single templates, eigen-based approaches can be used to capture the variability of objects, generating several templates that together could be used to detect objects in scenes. One example of such an approach has been applied in face recognition to yield the known Eigen-faces [8]. Usually, these algorithms require vast amounts of training examples in order to tune the system. On the contrary, mammals can learn many object categories from a few unsegmented examples. A lot of effort has been put in this direction to develop models that can handle large image databases and many object categories. Recently, methods inspired by biological vision have been shown to overcome the limitations template based techniques, which were partially successful for specific tasks [82], [63].

Histogram-based techniques have been developed for a number of applications, and have been shown to be powerful, partly because of their implied statistical nature, but also because they provide an easy way to achieve scale or rotation invariant descriptions.

One example of a successful method is that of Multidimensional Receptive Fields MRF histograms [81], which were developed to embed and simplify information within an image patch based on its properties. These high dimensional probability density functions solve the problem of correspondence, which can be used to compare and identify features.
2.1 Template Based Recognition

Template based object recognition is considered applicable outside the machine vision community as it does not deal with an object’s variation. Usually, this involves a patch used as a template where cross correlation outputs the most probable regions within an image that similar information might lie [33]. Face recognition methods still use template based methods such as PCA [71], ICA [6] and LDA [65]. PCA is derived from Karhunen-Loeve’s transformation. Given an n-dimensional vector representation of each face in a training set of images, Principal Component Analysis (PCA) tends to find a t-dimensional subspace whose basis vectors correspond to the maximum variance direction in the original image space [71], [1]. If the image elements are considered as random variables, the PCA basis vectors are defined as eigenvectors of the scatter matrix.

Independent Component Analysis (ICA) minimizes both second or higher order dependencies in the input data and attempts to find the basis along which the data (when projected onto them) are statistically independent. Bartlett et al. [6] provided two architectures of ICA: the statistically independent basis images and the factorial code representation.

Linear Discriminant Analysis (LDA) finds the vectors in the underlying space that best discriminate among classes [65]. For all samples of all classes, the between-class scatter matrix and the within-class scatter matrix are estimated. The goal is to maximize between-class variance while minimizing within-class variance.

2.2 Segmentation-Based

Segmentation is a grouping method which is usually applied to separate the foreground object from the background. One approach is through the use of k-means, which is applied on the image. An initial number of clusters is required for the algorithm to perform clustering. There is usually a random initialization of the clusters where a similarity measure performs a first pass on the pixels. The distance is the Euclidean similarity measure between a feature and a cluster center. The similarity is
typically based on a number of image characteristics such as pixel colour, intensity, texture, and location, or a weighted combination of these attributes. The number of clusters can be selected manually, randomly, or heuristically optimised. This algorithm clusters regions of the image but it is not guaranteed that one of the clusters would be the figure-ground or one of the objects within an image.

Another method to perform grouping is the region growing method. This method is initialised on image locations as an input to start the algorithm. These locations mark each of the objects to be segmented. The regions are iteratively grown by comparing all unassigned adjacent pixels with the initialised locations. The difference between a pixel’s intensity value and the region’s mean is typically used as similarity measure. The pixel with the smallest measured difference is assigned to the corresponding region. This process cycles through until all pixels are assigned to a region. Initialised (Seeded) region growing requires starting points (seeds) as auxiliary input. The segmentation results are dependent on the choice of the initial points. Noise in the image can cause the initialisation points to be poorly placed. Uninitialised (Unseeded) region growing is a modified algorithm that does not require explicit auxiliary starting points. It starts off with a single region where the pixel chosen does not significantly influence final segmentation. At each iteration, the neighbouring pixels are considered in the same way as initialised region growing. It differs from seeded region growing in that if the minimum average is less than a predefined threshold $T$ then it is added to another respective region. Otherwise the pixel is considered significantly different from all current assigned regions and a new region is initialised.

The split and merge method starts at top parent node of the tree that represents the whole image. In case of a non-uniform (not homogeneous) region is found, it is subdivided into four subsequent-squares (the splitting process). The four squares are examined, if there are homogeneous, they can be merged as several connected components (the merging process). The node in the tree is a segmented node where this process continues recursively until no further splits or merges are possible [51]. When a unique data structure is involved in the implementation of the algorithm, the method of [51], can reduce the complexity to $O(n \log n)$, obtaining an optimal algorithm for the method.

The watershed transformation is applied on top of the gradient mag-
The magnitude of an image which is further treated as a topographic surface [24]. High gradient magnitude locations are considered start off points of the algorithm, which are actually edges of regions. The algorithm is thought as water placed on top of the edges which is moved by the gravity downhill to lower gradient magnitudes. Low magnitude regions form drain regions of the water downhill movement which are considered the desired segments.

Graph partitioning methods have been used also for foreground segmentation. In these methods, the image is modelled as a weighted, undirected graph [14]. Usually a pixel or a group of pixels are clustered together by spectral clustering approaches with nodes being the segments and the edges of the graph holding the pixels into a segment. The variability of the nodes in the graph are examined by an optimisation approach until two segments are formed which constitute the foreground and the background. Each partition of the nodes is generated from algorithms of randomisation and optimisation which consider an object segment in the image [14], [72].

2.3 Multilayer Learning Based

There is an abundance of machine learning methodologies that attempt to perform object recognition tasks. The majority of these approaches are organised into layers of learning modules where each learns some property of the output of the previous layer that is a discriminant to inputed data. For instance Deep Boltzmann machines [80] are organised into layers where each node within a layer is a hidden Boltzmann machine. Also, a similar organization is followed by Hidden Markov Fields [2] and their variant Conditional Random Fields [59]. The neural network community pioneered hierarchically organised approaches with the perceptron [50] being one of the earliest implementation. Generally, these organised layers even attempt to imitate neuronal organisation in mammals; they do not usually perform efficient object recognition. Despite these approaches, there are methods which perform class separation on the feature space of vectors conveying specific attributes. Even though the methodologies of margin classifiers have been applied as multilayer approaches, these have failed compared to simple counterparts. The success of the margin classifiers comes from the class separation techniques
in the feature space of vectors. Specifically, Adaboost [42] and Support Vector machines [23] are the most widespread implementations of margin classifiers, with state of the art performance.

2.4 Scale Invariant Feature Transform

One of the key requirements for real-world object recognition is scale invariance, which enables comparisons of visual information between various regions of interest captured at different scales or levels of zoom. A significant disadvantage to template matching is that the constant template size limits the matching to the equivalent ROI. Without a means of compensating for changes in camera-object distance, incorrect matches, false positives, or false negatives can occur. Generally, it is thought that the SIFT descriptor has improved the performance and scalability of retrieval algorithms [63].

T. Lindeberg [60] proposed an automated scale selection method in Gaussian scale space where the approximate centers of blobs in scale space are detected as local extrema. The SIFT keypoint detector exploits these locations capturing information on the location of the blobs (in scale space) and the region around the keypoint which characterizes the dominant spatial frequency of a candidate local structure. In SIFT, scale space construction may be seen as a computational approximation to responses of retinal ganglion cells of different sizes to a grey-scale visual input and the keypoints as focus of visual attention. Another approach, biologically inspired, is the saliency detector [54] where local entropy is used as a measure for selecting keypoints. This method was not investigated further because it is already included in the study of [68].

2.4.1 Scale Selection

A proportion of ganglion cells in the retina will consist of receptive fields which display center excitation and surround inhibition. A spatial filter with similar properties would respond to regions of illumination contrast, generating strong responses in these areas [55]. An approximation to the spatial receptive field of a ganglion cell can be obtained by the subtraction of two Gaussian filters (Difference of Gaussians DoG filter). The DoG filter is comprised of two Gaussians of different $\sigma$ values. By changing the $\sigma$ values, the properties of the DoG can be altered, including spatial
frequency and bandwidth. The term *spatial frequency* refers to the rate of illumination variation along the image spatial location. The value of $\sigma$ plays an important role in tuning the spatial frequency selectivity where large $\sigma$ values correspond to large kernel sizes with an increased proportion of low spatial frequencies.

![Diagram of DoG scale space](image)

**Figure 2.1:** The DoG scale space is obtained by blurring an image repeatedly with Gaussians and computing the difference between these blurred representations. The image is resized to make the spanning of several octaves efficient. By halving the size of the image, a drop of 1 octave is achieved in the spatial frequency domain. Reproduced from [63].

The repeated application of Gaussian blurring in this way generates a representation of the image that is known as a *scale space representation*, in which the scale of the image and the scale (size) of the filter directly emphasise a band of spatial frequencies present in the image. The repeated application of a Gaussian blurring kernel within one octave is analogous to applying a series of Gaussian filters of different widths to the image – a filter bank. After the filter bank has been configured, the outputs from the different filters are subtracted to generate a DoG space representation. At this point, one can draw a rough analogy between the retinal ganglion receptive field. Within this scale space, blobs are detected by locating singularities using a max operator across scale space.
as shown in Figure 2.1. The result provides information about stable local regions with associated dominant frequency band.

### 2.4.2 SIFT Descriptor Construction

The computed scale space is subjected to further processing to determine the orientations of the frequencies. Lowe suggested [63] that this was a process that was similar to the the actions of complex cells in V1. In fact, this is only partially true, as SIFT is phase selective, generating responses similar to a simple cell in V1 that is selective to spatial translation. This is because, as illustrated in Figure 2.2, spatial derivative operators are applied to estimate spatial gradients within a given scale, or equivalently, a spatial frequency band. The result is illustrated in Figure 2.2, where this region is sampled using histograms of oriented gradients [63].

![Image](image.png)

**Figure 2.2:** A keypoint is detected in scale space as previously described. This point is remapped to the original image to calculate the image gradients. The gradients are computed by derivative operators and weighted by an isotropic Gaussian window represented by the blue circle above. This region is split into 4 quadrants where each is sampled using a histogram of oriented gradients (HOG). Each bin of the HOG represents a gradient orientation, and there are 8 bins in total. Having acquired 4 HOGs as illustrated on left side of figure, histograms are generated which are incorporated into the SIFT patch descriptor. The figure illustrates only 4 sub-patches instead of 16 which are the implementation. Reproduced from [63].

The arrangement of spatial gradients over a patch is captured by histograms which are typically comprised of 8 orientation bins. The area around the keypoint that will be included is determined by the scale of the keypoint. That area is represented with the histogram of gradients
in order to establish a principal orientation of the patch which also is associated with the keypoint’s location and scale. Using this information, the whole sampling area is rotated according to the dominant orientation and resampled. The rotated patch is divided into 16 smaller rectangular regions where each is again sampled using the histogram of oriented gradients. This yields 16 histograms which are appended in a single 1D array forming a 128 dimensional vector that describes a local image patch. This 128 dimensional vector is known as the SIFT descriptor.

2.5 HMAX a Computational Model of Visual Cortex

Hierarchical Model And X (HMAX) is a neural computational model that is intended to simulate biological object recognition. The model is organized into layers comprised of simple and complex cells that lie in the visual cortex. A battery of even symmetric Gabor filters is applied to imitate V1 simple cells selective to 4 orientations [82]. The approximation of the shift-invariance provided by non-linear processing is achieved by max pooling; one can see this as converting a “simple cell” response into a “complex cell” response.

In the HMAX model, there are two main group of cells that approximate V1 (S1,C1) and V2 (S2,C2) tunings. The (S1,C1) are simulated by even symmetric Gabor functions for V1 tunings. The (S2,C2) group responses of V1 (S1,C1) to approximate the grouping occurred in V2. The higher layer accumulates the responses of V2 to obtain a similar sensitivity to those observed in V4 and IT [19].

2.6 Approximating Responses in Striate Area with Gabor Filters

In image processing Gabor, or Gabor-like filters have been used for a huge variety of vision tasks, including low-level feature extraction in robot vision [95], biometric applications [30], including fingerprint analysis [49] and face recognition [62],[104],[101], and texture analysis [34]. The Gabor space is generally overcomplete: that is, each pixel in the image is mapped to more than one Gabor output. The degree of overcompleteness depends
Figure 2.3: HMAX model utilizes a series of layers of simple and complex cells to perform object recognition. Starting from the bottom, each layer accumulates information from the previous layer using weighted sum and max pooling as indicated. The top layer approximates a response of V4 and inferotemporal cortical area (IT)[82].

on how the Gabor parameter space is sampled. A benefit to the use of an overcomplete representation for inference is that the descriptions of local image structure produced by such a mapping has quasi-invariant properties, stabilizing a descriptor’s variability across tilt, illumination, rotation and scale. As in the case of SIFT, for example, quasi-invariance in scale is obtained by selecting the appropriate scale of Gabor wavelet to transform a region into Gabor’s phase and orientation space. Also, the selection of the best tuned wavelet at each point in space provides a frame of angular reference which results in quasi-invariance to in-plane orientation.

However, perhaps the most important property of Gabor filtering, and other approaches that are based on combining symmetric and anti-symmetric filter responses of equivalent spatial support, is that one may obtain a description of the local image structure that is either phase invariant, or which is phase selective; this is very a similar idea to the use of the Discrete Fourier Transform (DFT), in which both magnitude or
phase can be extracted, depending on the application.

Unfortunately, Gabor methods have not been applied to large-scale, real-world object recognition despite successes in the areas identified above.

2.7 Conclusions

If we consider the three main biologically inspired methods, briefly described in this chapter, it is clear that the method that has been successful, and adopted by the computer vision community to the greatest extent is the SIFT descriptor. This is surprising, because SIFT is the least biologically plausible, based on the fact that it only uses simple multiscale gradient fields to analyse spatial structure. What are the main reasons behind the success of the SIFT descriptor for object recognition?

SIFT is considered to be very easy to use, and is computationally very efficient. There are two main reasons – each image is decomposed into a collection of fixed-length descriptors, which could represent patches of very different size. This may be thought of as a focus-of-attention mechanism, but it is applied in such a way as to produce very convenient descriptors. The second major reason is that the descriptors are only calculated around stable keypoints which include a local scale and dominant orientation estimate.

Neither HMAX nor Gabor jet approaches have, to date, included a robust, compact descriptor at only key locations, preventing the highly scalable and accurate performance of SIFT. At the same time, it is thought that SIFT descriptors do not work well for face recognition, where Gabor jets excel [1]. This presents a problem to systems that are designed for analysing all images, irrespective of content: it suggests that we would have to use different front-ends depending on the image type.

If, instead, a keypoint-type approach could be added to Gabor-base representations, the same front-end (i.e. multiscale spatial filters) could be used for both face recognition and object recognition, provided that the performance is at least as good as SIFT and its variants.
In the next Chapter, a flexible front-end is designed that allows Gabor-like responses to be optimised for performance, and for keypoint detection and scale-estimation. This is used in some of the subsequent Chapters.
Chapter 3

Filter Bank Design

3.1 Introduction

The majority of intensity based object recognition frameworks employ, in some way, a set of filters to examine spatial gradients at multiple image resolutions. This helps to achieve a scalable image representation where it is proven that using gradient representations yields better performance than any other processing method at this stage [81]. Generally, gradient fields have become the front end ingredient for object recognition. This is partly due to low level invariant properties that convey such as tolerance to global illumination changes. Furthermore, the filter design is further analysed for a suitable front end such as bandpass filters to acquire gradient fields. Besides gradient fields an alternative approach is the use of local entropy [54] for creating a feature space, even though local entropy is not a successful approach to image keypoint detection [68].

3.2 Related Work

The image processing community has found that log-normal energy responses provide good performance across a variety of elementary computer vision and pattern recognition tasks such as texture segmentation [48], image denoising [40],[41], and biometrics [1],[4] including fingerprint analysis [97]. In addition, biological evidence shows that the early visual processing stage in certain cells of mammalian visual cortex can be approximated by linear operators with spatial transfer functions described by skewed Gaussian energies [39] (log-normal distributions).

There are many scale space approaches that attempt to accurately de-
compose an image into spatial frequency bands. A Gaussian scale space is a common example of a well studied space in terms of scale properties where derivative kernels of normal distributed envelopes produce a multi-resolution representation. Basic properties of such scale space have been studied by [60] to tackle scale invariance of local structure. Another scale space derived by Poisson kernels (also known as \( \alpha \) scale space) was introduced and studied by [37]. The advantage of using Poisson kernel is mainly due to its distribution when comparing it with a normal distribution. The tail of the Poisson distribution makes the envelope of the filter decay more slowly, comparing to a normal distribution (Gaussian envelope) which decays faster. Also, skewed distributions such as Poisson, Gamma and log-normal are highly correlated in terms of shape: one can obtain similar skewed envelopes by changing the parameters of these distributions.

Similar skewed envelopes are applied to build 2D log-normal filters in the Fourier domain to create a log-normal scale space. The main reason for selecting the log-normal distributions is that all the properties of those distributions have exponential terms to describe their properties which under a Fourier transform the majority of their properties are transferred intact to the spatial domain. In addition, these are more manoeuvrable when one designs in Fourier space and their tails approximate the power decay of the Fourier spectrum of natural scenes.

The successful SIFT implementation uses Difference of Gaussian filters by [63], where the differences in response of filtering an image with isotropic Gaussian filters of varying widths is used to decompose an image into several radial bands of spatial frequencies. The DoG filters are applied every third of a scale octave \( \frac{1}{2^3} \) to provide an accurate scale estimate. This approach was suggested as a suitable automatic scale detection scheme by [60] to achieve scale invariant features which are usually repeatable under various image scale changes. A common method to detect such features is by detecting local extrema along image space and across scales. Further regression and curve fitting methods are employed to predict the precise position of extrema in scale-space. A typical method is 3D quadratic curve fitting to acquire sub-pixel and between-scale accuracy [63].

The purpose of the work described in this Chapter is to equip a set of feature detectors, similar to a Gabor jet, with keypoint detection and
scale estimation. The design of the front-end filter set is essential for detecting scale invariant features. Enabling scale-estimation is a fundamental part of keypoint-type approaches.

Our filter design is applied in the Fourier domain to simplify and reduce the computational load of creating a complex filter bank. Furthermore, the assessment and setting of the frame properties of an overcomplete wavelet dictionary is feasible in discrete Fourier space using the energy distribution of the desired filters. For example, we can estimate frame constants [64], and use this as a method to optimize the completeness and redundancy of the filters. In addition an optimisation method is taken to create sparse Fourier representations. The effect of some tuning was found (in Chapter 4) to yield better keypoint detection performance.

3.3 Design of the Filter Bank

A filter bank was designed with respect to balanced wavelet outputs of different orientations and scales of an image an optimisation process is applied to estimate the optimal radial and angular spacing in the Fourier domain. Although the bandwidth of the filter can be arbitrary large, there are restrictions due to their spatial width if applied as kernels\(^1\). The log-normal filters can have small spatial width from approximately 0.7 up to 2.5 octaves of bandwidth. In image reconstruction, as shown in [41], such filters are applied up to the 5th scale in an octave wise spacing, accurately reconstructing the image.

In this work, an overcomplete wavelet design is adopted [40] using the following “mother” kernel equation to optimize the frequency spacing of the filters. The authors [40] make use of 7 scales to perform image denoising. Because image categorisation is not considered robust using existing algorithms, this work pursues the effect of tuning bandpass filters towards better categorisation; hence the spatial frequency bands that give better discrimination for categorisation are explored (see Chapter 5 for more details).

\[
\hat{\Psi} = \exp\left(-\frac{(\ln\rho - \mu_\rho)^2}{2\sigma_\rho^2}\right) \exp\left(-\frac{(\theta - \mu_\theta)^2}{2\sigma_\theta^2}\right)
\]

\(^1http://www.csse.uwa.edu.au/~pk/research/matlabfns/PhaseCongruency/Docs/convexplo.html
Equation (3.1) describes the log-normal energy profile of a spatial wavelet in 2D Fourier where $\mu_\rho$ is typically the geometric mean in this distribution (with $\rho$ the radial support for the frequencies) and can be estimated by the logarithm of the mode. The mode corresponds to the central frequency of the filter, which is used as a reference point to set the radial spacing. Also, the denominator $2\sigma^2_\rho$ represents the geometric standard deviation of the first term. This parameter modulates the radial bandwidth of the filter, and its value is obtained by the optimization process in equation (3.3). The second term of the wavelet’s energy ($||\Psi|| = \hat{\Psi}$) defines the angular spread of the filter where $\mu_\theta$ is the mean of a normal distribution (with $\theta$ the angular support of the frequencies) which is used to tune the filters on 8 orientations in half space. The $2\sigma^2_\theta$ defines the angular spread which is also acquired in the optimization process.

### 3.4 Filter Bank Optimization

The wavelet dictionary is comprised of 5 scales and 8 orientations at each scale where the constituted frame is optimized to the optimized redundancy level using frame constants. A redundant Fourier representation of 40 basis vectors creates a 2D vector $\psi_1, \ldots, \psi_{40}$ frame basis (with $\psi_1$ representing one of the 40 different distributions produced by $\hat{\Psi}$), where the vectors are projected. A flat (linearised) frame produces a filter bank for accurate frequency estimation and interpolated position and steering [64]. Several rotated and scaled versions of Equation (3.1) constitute a wavelet dictionary. Each term (wavelet) in the dictionary symbolizes a member of the basic set or the basis vectors (or Dictionary of wavelets) $D = \{\psi_1, \ldots, \psi_{40}\}$ (also referred as wavelet dictionary). All these vectors can be expressed in linear dependent form in (3.2) under a basis or a frame of vectors which will be further linearised (flattened).

The dictionary $D$ should be optimised for signals, represented by a signal vector $f$ of the same length as each of the dictionary terms, $\psi_i$; this is done through the frame constants, $A$ and $B$ such that:

$$A\|f\|_2^2 \leq \sum_{w=1}^{40} \langle f, \psi_w \rangle^2 \leq B\|f\|_2^2 \quad (3.2)$$

The variables $A$ and $B$ are the minimum and maximum values of the squared norm filter output in the frequency domain. Hence, $A$ and $B$
define the borders of the frame where tighter frames lead to improved numerical closeness of the signal reconstruction under the dual frame. During the optimization process, Equation (3.3) is applied using Mal- lat’s [64] sparse-dictionary interpretation of frame optimisation to obtain the best possible 2D spacing of the filters in half Fourier space. Frame constants $A$ and $B$ in expression (3.2) can be converged to its other ($A \approx B$) by minimizing a Lagrangian with a squared loss function of all $\hat{\Psi}$ over their mean with a lower basis (second term) to keep $B$ close to $A$.

![Figure 3.1](image)

**Figure 3.1:** This figure illustrates an unoptimised frame. The purple curves represent the log-normal distributions in Fourier space. The blue line corresponds to the unoptimised frame.

There are three main performance advantages in optimising in this way. Firstly, the closeness of the frame constants ($A \approx B$) yields a balanced filter bank where linear combinations of the filters can produce accurate estimates in terms of orientation and frequency estimation. Secondly, the optimised wavelet dictionary has improved noise tolerance compared to an unoptimized version, as will be shown in the next chapter. Finally, the balanced redundant dictionary in the Fourier domain yields a sparse response in the spatial image domain, due to band-limited spatial
frequency representation comparing to the original image sampling rate.

\[
\hat{\mathcal{L}}(\mu_0, \sigma_\rho, \sigma_\theta) = \int_0^{1/2} \left( \|\psi_{r,j}(\rho_\omega)\|_2^2 - \mu_{\hat{\Psi}} \right) d\rho_\omega + \lambda \mu_{\hat{\Psi}} \quad (3.3)
\]

\[
\mu_{\hat{\Psi}} = 2 \int_0^{1/2} \|\psi_{r,j}(\rho_\omega)\|_2^2 d\rho_\omega \quad (3.4)
\]

The average squared energies set the target value \(\mu_{\hat{\Psi}}\) that the frame (defined by: \(\|\psi_{r,j}(\rho_\omega)\|_2^2\)) constants must converge to by minimizing the loss function of Equation (3.3). In other words, the mean value represents the ideal pursued tight frame. The second term bounds the upper level of the dynamic range (the level of the higher basis constrained by \(\lambda\)) forcing it to be approximately uniform in the frequency domain. Enforcing the filter responses to have uniform distribution aids the regulation to the filter responses which improves orientation and scale estimation (see Chapter 4). Throughout our experiments, precise scale estimates are found to be better if the frame (A and B bounds) converges to level of bias around 1.56 (frame value see Figure 3.1). Lower values than 1.56 reduce the ability of the dictionary to provide representative scale estimates (see next chapter). A uniform distribution is the optimal configuration of the filters’ frame with radial bandwidth 1.47 octaves and angular bandwidth 0.54 octaves. The central radial frequencies are set on each scale octave covering more than 95% of the 2D frequency spectrum. The angular spacing is comprised of 8 orientations in half space providing the choice of phase invariant responses or phase selective. The parameter settings described above were obtained after performing tests on interest point detection as described in Chapter 4.

A learned, linearised (flattened) basis is highlighted on Figure 3.2, where the scale octave radial spacing is optimized under a range of radial bandwidths to yield the illustrated frame. Similarly, the angular bandwidth of 8 directional filters is optimized through Equation (3.3). Both frame bases are usually represented by the second term of Equation (3.3) and are equalized to produce a flat 2D frame across the 2D Fourier domain. The learned terms provide a balanced wavelet dictionary in which linear combinations of the dictionary members can yield sub-scale and sub-orientation (orientations estimates lower than the set filter orientations) estimates.
Figure 3.2: The red curves show the optimized spacing where no oscillations appear along the curves. The final sigma values are: $\sigma_\theta = 0.346$, $\sigma_\rho = 0.612$ used to the final configuration of the wavelet dictionary.

As shown in Figure 3.3, 5 rescaled and 8 rotated versions of log-normal distributions cover half of Fourier space. The choice of half space enables us to gather complex responses from the cosine and sin parts from the Fourier transform, by which one can further compute the magnitude and phase responses. The filters are arranged in the Fourier domain in similar fashion of a polar coordinate system where towards the center the spatial frequencies become smaller. The maximum frequency starts at 0.5 cycles per pixel and ends at the origin which is zero. A positive direction is considered anti-clockwise with range $(-\pi, \pi)$. Each scaled and rotated version of a filter is independently applied to the transformed image and projected back to the spatial domain by inverse Fourier transform. Consequently, 40 channels are gathered and classified according to their orientation and scale for further use.
Figure 3.3: A Fourier representation of the optimized filter set. All scales are independently mapped to collect the spatial responses. Above, for illustration purposes, each filter is normalised to have a peak amplitude of 1, and the filters are combined by a max() projection onto the \( u_y, u_x \) Fourier plane.

3.5 Local Phase Encoding

Complex filter responses can be thought of as complex vectors with real and imaginary components. Often, these filter types are orthogonal, which can occasionally be normalized to yield an orthonormal basis [64]. This means that the real and imaginary parts have a \( \frac{\pi}{2} \) angle offset from each other. This filter definition enables us to treat the filter pairs as complex vectors. Hence, the magnitude of the complex output provides shift invariant properties to the domain that the complex filters have been applied. In our case, the shift invariance becomes the oriented magnitude responses of local structure or image primitives such as directional impulses and step responses or any variation of these two types. These are usually transformed into one non-negative response.

Similar, local structure types can be estimated by the use of the phase component. A phase estimate can be computed using the arc of tangent (inverse tangent function) from the real and imaginary part. Figure 3.4 illustrates the type of local structure that can be encoded by the phase.
Figure 3.4: A typical one dimensional signal “unravels” phase properties of a 1-dimensional signal from a 2D phase-space to the spatial domain. The horizontal axis on the top two plots show pixel locations and the $y$ axis shows pixel intensity and phase angle output respectively.

In particular, common locations of interest are highlighted (red dots) showing the correspondence of spatial location to the phase estimate. The top plot in Figure 3.4 presents a signal composed of two types of impulses and edges, respectively. The corresponding phase response is shown in the middle plot. It can be seen that the phase signature is unique for each type of structure. The polar plots suggest that the complex pair outputs are finely tuned to certain structures even though a small position shift can produce dramatic measurement change. Hence, the phase behaviour may be utilized in the following ways: (a) phase response can be used for accurate keypoint localization leading to subpixel accuracy [57]; (b) similar local structures can be identified by the use of phase.
The choice of filters plays an important role in the overall performance of object recognition. Thus, careful design on the filter bank must be given to guarantee that the best of these filters can be utilized. An optimisation method will be proposed to create balanced filter outputs aimed at achieving stable behaviour of the filters, especially for scale detection and orientation estimation processes, which are discussed in the following chapters. Specifically, a novel optimization approach was presented based on the frame properties of the filters, aiming to tune filter responses for improved scale and orientation estimation. The magnitude of the complex filter response approximates the shift invariant behaviour of biological complex cells in V1. The phase components of the filter responses can indicate local structure appearance, and individually ap-
proximate simple cell responses in V1, which are selective to local image phase.
Chapter 4

Scale Estimates

An isotropic scale space representation of an image is a decomposition of its spatial frequency spectrum into bands of different radial frequency spacing. These bands usually convey different aspects of an object’s structures. Gaussian scale space has a different mathematical root relative to other multiresolution approaches [60]. Common approaches to construct a scale space include Gaussian kernels, including Laplacian of Gaussians, Poisson kernels [37], Gabor jets [56] etc. An easily made observation is that blob like structures emerge in many image scale space decompositions. Searching over scale, one will typically observe that blobs have specific lifetimes in scale. Thinking in terms of frequencies, blobs emerge when the band of the applied filter matches the band of spatial frequencies which constitute a particular structure. Sometimes, small blobs coalesce into a larger blob which exhibits higher lifetime and amplitude in scale space. An explanation of this phenomenon might be due to the fact that band limited filters can hold frequencies from adjacent scales. Additionally, neighbouring blobs also merge if this region can be better matched by a coarser scale, implying lower wavelet frequency if the scale space is generated by wavelet analysis.

In this chapter, the use of keypoint methods based on scale space decomposition is addressed. Good selection of the keypoints can lead to more efficient performance in object recognition. Many alternative methods have been proposed, in which the basic principle is based on 3D maxima [60] for keypoint detection.
Figure 4.1: This Figure illustrates the Gaussian scale space of a simple one-dimensional signal. A line from the middle of the height of the image which contains rectangular white boxes has been extracted. The blue line highlights the extracted one dimensional signal. The output of increasing size filter produces the scale space in this example. It is clearly shown that local maxima appear along the scale in this space which often are described as blobs. For a 2D signal (image) a 3D space is formed where the responses around the maxima in this space form spherical regions also known as blobs.

4.1 Scale selection in Log-Normal Scale Space

Typically, in images, there are many object size variations which simple template matching cannot efficiently tackle. A scale space decomposition handles the scale variability by assigning image regions to a scale estimate. One method to identify these regions is maxima detection over scales along the image. This provides unique scale measurements of local structure. For instance, symmetric and antisymmetric filters applied to a region can roughly describe the local structure appearance. Scale estimation could also determine the sizes of these structures. Alternatively, each local structure has an implicit size, and a scale space decomposition
can encode the size of a local feature. Many approaches [63], [7], [27] employ the amplitude of spatial derivatives in a multiresolution space to estimate the scale of local structures. There are two ways to acquire the scale space of an image: either rescale the filter or downsample the image.

In this Chapter, a collection of filters applied to yield a log-normal scale space. The optimization of these filters was explained in the previous chapter: the filters are optimized over both radial frequency and angular frequency. Radial frequency can be correlated with scale, as radial frequencies usually correspond to structure size. Aiming to focus on the scales (radial bands in Fourier domain), two methods are suggested to compose angular bands into a scale space representation.

### 4.2 Scale from Directional Filters

Once the complex filter outputs have been generated, two different methods were compared to produce stable points in the image that could be used to assign an intrinsic scale and location, similar to keypoint methods. The two methods are described in Equations (4.1) and (4.2). Scale estimation was performed with the aid of Gaussian calibration blobs (see Figure 4.2), where $|C(x, y, o, k)|$ denotes the magnitude of the complex filter field in direction $o$ at discrete scale $k$.

$$V_{ss}(x, y, k) = \left( \prod_{o=1}^{N_o} |C(o, x, y, k)| \right)^{\frac{1}{N_o}}$$  \hspace{1cm} (4.1)
Figure 4.3: On top, the original image taken from the Pascal VOC 2011 class aeroplane. The left column shows a series of the $T_{ss}$ response at the 2nd, 3rd, and 4th scales. The right column shows different scales of $V_{ss}$. 
The two methods convert raw filter outputs into a measurement that can be related spatial scale. Equation (4.1) can be viewed as a geometric mean; or simply a normalized measure of ($N_o=8$) orientations. This operator tends to produce large responses on edge junctions. Alternatively, Equation (4.2) sums the magnitude of complex outputs, which may be seen as the trace of a tensor [10] representing the oriented directional filter magnitudes.

Local extrema in ($x, y, k$) are then searched across either $V_{ss}$ or $T_{ss}$ to define keypoints, and to assign scales to them. A first pass detects points that satisfy 3D maxima conditions; a simple scale localization approach is used to estimate sub-scale accuracy:

$$
\sigma_i = \frac{\sum_{k=1}^K V_{ss}(x_i, y_i, k) \sigma_0 \cdot 2^{k-1}}{\sum_{k=1}^K V_{ss}(x_i, y_i, k)}
$$

(4.3)

An optimization approach can also be used to reduce the estimation bias due to the open-interval sampling of scale-space; for example one could use a third-order polynomial regression to learn the correction term using inputs of Figure 4.2 at the lowest noise level. The accuracy of keypoint detection and scale estimation of both methods is evaluated in Section (6.1) using different levels of noise added to the Gaussian blurred impulse functions with a range of scale values, and in the presence of noise.

### 4.3 Optimizing Scale Estimates

Keypoints, detected using the method described above, were evaluated for scale and location accuracy against other interest point detectors using homographies from a standard benchmark database [68], which contains images of varying quality. Overlapping keypoints were penalized; without such a penalty, a dense, regular grid sampling of keypoints would lead to the best repeatability measure, which defeats the scalability of the keypoint approach: provided that keypoints are stable in the object recognition context, overlapping keypoints add no great benefit to the descriptor feature space. Even though Equation (4.3) provides fractional
scale estimates, these diverge from the ground truth for very low and very high scales. This can be more easily understood by referring to the flat frame in Figure 3.2, which shows the radial components of the filters; the frame declines from a plateau near to both ends of the log-normal distributions. A validation of this claim is provided by Figure 4.4, where scale estimates show similar behaviour at the frame borders in Figure 3.2. To remove this bias, a curve fitting approach was adopted to learn the deviation of the estimates in those areas. The third order polynomial curve fitting is applied.

\[ y_i = \alpha_0 + \alpha_1 x_i + \alpha_2 x_i^2 + \alpha_3 x_i^3 + \epsilon_i \quad (4.4) \]

\[ Y = [y_1 \ y_2 \ \ldots \ y_i]^T \quad (4.5) \]

The \( Y \) in our problem is the output we wish to learn, which is the error correction of the scale estimates with respect to ground truth, measured in a finite domain. The ground truth is specific scale value set by Gaussian distributions as shown in Figure 4.2. The input \( X \) is a Vandermonde matrix where each scale estimate is a different index of \( x_i \).

\[ X = \begin{bmatrix}
1 & x_1 & x_1^2 & x_1^3 \\
\vdots & \vdots & \vdots & \vdots \\
1 & x_i & x_i^2 & x_i^3
\end{bmatrix} \quad (4.6) \]

Hence, our objective is defined as feeding scale estimates into a function which outputs the actual error of each scale value comparing to the ground truth. The Gaussian spatial distributions in Figure 4.2 defines the ground truth, which is octave wise scale spacing to test our filter bank responses. This enables us to establish a realistic scale detection schemes that allows to benchmark the scale accuracy against other methods such as DoG, Harris Laplace and Hessian Laplace.

The optimum estimation (ground truth) lies on:

\[ y = \alpha x + \beta \quad (4.7) \]

Equation (4.7) determines a ground truth (true values of Gaussian width) function, with \( \alpha = \frac{4}{5} \) and \( \beta = 1 \), which is highlighted as a red dashed
Figure 4.4: This graph illustrates the deviation of the ground truth (red dashed diagonal line), from the estimated scales using $T_{ss}$ (sum) and $V_{ss}$ (product) to reconstruct scales from the steered log normal responses.

diagonal line in Figure 4.4. As previously discussed, $Y$ defines the difference of scale estimates $\hat{\sigma}_i$ from the actual scale $g_i$ values. Next, the $X$ is defined as the generated scale estimate using Equation (4.3).

\begin{align*}
Y &= g_i - \hat{\sigma}_i \\
X &= \hat{\sigma}_i
\end{align*}

An polynomial function receives scale estimates and yields error values as illustrated in Figure 4.5. This third order polynomial function conveys the coefficients that minimise this error output by deploying polynomial curve fitting.

\begin{align*}
A &= [\alpha_0 \quad \alpha_1 \quad \alpha_2 \quad \alpha_3] \\
A &= (X^TX)^{-1}X^TY
\end{align*}

Specifically, Equation (4.11) provides an approach to acquire such a so-
olution by generating the \( A \) coefficients that form the shape of the errors’ curve. These coefficients are further used in Equation (4.4) to reduce the error bias in the scale estimates.

![Error Prediction and Correction](image)

**Figure 4.5:** This graph presents the produced error of the estimated scale relative to the ground truth. Also the learned curve is highlighted, showing only small deviation from the best possible correction.

Finally, noisy error rates with 0.01\% Gaussian noise are added to the test images (Gaussian distributions) to learn the corrective \( A \) matrix (4.10) for the scale estimates. After applying the previously described procedure, the scale accuracy of this approach is compared against other keypoint detection approaches. Using the same scale spacing of the Gaussian configurations of Figure 4.2, the outputs of the detector are mapped onto a finite scale space to address the localization and scale accuracy. The candidate detectors are MSER, Harris-Laplace, Hessian-Laplace and SIFT (DoG) detector. An example of the finite scale space is illustrated in Figure 4.2. The results of the detectors, including the correction in our scale estimates for both sum and product are presented in Figure 4.6. In the first experiment, Harris-Laplace is tested for generating very unstable estimates (see Figure 4.6). Observing the generated curve, it is noticeable that high deviation of the estimates occurs in the presence
of additive noise. In addition, there is a general trend to follow similar behaviour as the ground truth slopes. There is a constant bias in the curve, which can be easily removed. The Hessian-Laplace case shows similar characteristics to Harris-Laplace, but its variance is higher among all comparisons, making it the most unreliable. Both detectors appear to have a constant bias from ground truth which can be interpreted as a scale offset at the finer scale estimates. This probably depends on the implementation of the detector, itself rather its theoretical formulation.
Figure 4.6: The diagonal red lines show the optimal scale target in log₂ space. The y axis shows the estimated scale using 10 intervals of random Gaussian noise. The error bars indicate the average standard deviation for the [0.1%, 1%, 10%, 30%] noise levels.

Next, the MSER detector shows more stable scale estimates, although there is a bias as well. This bias is likely to be due to stable region estimation of the MSER algorithm, where the added noise distorts the Gaussian regions, leading to divergence of the scale estimates. Despite the easily removable bias, the estimated curves are much more oscillatory, making
its use discouraging. The scale estimates of the SIFT detector closely follows the ground truth making it reliable. But despite the desirable trend of the SIFT’s scale estimates, the added noise increases the variance of the scale estimates, especially at the large scales. Finally, the scale estimates of the sum ($T_{ss}$) and product ($V_{ss}$) show low variance comparing to the other methods. Both follow the trend of the ground truth with a minor offset in the actual values. A side effect of the added noise on the estimates is clearly seen at large scales, where the estimates diverge significantly from the true values. This is probably due to the added noise which is perturbing the Gaussian shapes, causing detectors to fail in some cases.

4.4 Keypoint Evaluation on Benchmark Sets

The keypoint detection was tested on an evaluation protocol set by K.Mikolajczyk, et al. [68], which is widely used for evaluation of keypoint detectors. The protocol is comprised of matching criteria to assess the repeatability of the keypoints against a test set\(^1\). This database includes six different image sequences with a different type of transformation applied to each. Specifically, starting as illustrated in the Figure, the bark sequence (a) involves in plane scale and rotation changes of a series of bark texture image. Next, the “bikes” test sequence contains blurring variations with sigma values $2 - 6$ being used to imitate scale changes. The “boat” sequence includes a blend of out-of-plane scale and rotation changes, as well as planar rotation. In this case, a rough scale estimate varies between $1.12 - 2.8$, with planar rotation and non-planar rotation, ranging $0 - 90$ degrees. In the “graffiti” sequence, a slant angle or non-planar tilting up to 60 degrees with foreshortening is introduced to accommodate the same visual information in the frame. The “car” sequence shows a photometric transformation with decreasing light conditions using the camera’s aperture. Finally, the “UBC” sequence demonstrates increasing JPEG compression of up to 40%. The image sequences are related by homographies, so that a comparison can be made with average image resolutions at approximately $800 \times 640$ pixels.

\(^1\text{http://www.robots.ox.ac.uk/~vgg/research/affine/}\)
different images of the same scene must satisfy two repeatability criteria:

\[ |x - Hx'| \leq 1.4 \]  
\[ 0.8|\lambda_1| \leq \left( \frac{\sigma_i}{\sigma_j} \right) \leq 1.2|\lambda_1| \]

where \( x = (x, y, 1)^T \) and \( x' \) and \( \lambda_1 \) is the principal (complex) eigenvalue of the upper left \( 2 \times 2 \) submatrix of the pair homography matrix, \( H \). Equation (4.12) considers keypoints that pass the position criteria as those that lie within 1.4 pixels of each other [69]. Absolute, rather than scale-relative, accuracy is used because object recognition and detec-
tion may sometimes involve the use of “tight” specifications on keypoint 
homographies, to determine valid matches. The ratio between the esti-
mated scales of candidate match pairs must also be close to the global 
scale change between the original and transformed image. Specifically, 
the scale estimate of the candidate matching keypoint in a transformed 
image must be within 20% of the reference global scale change between 
reference and transformed images [69].

![Graph](image.png)

**Figure 4.8:** The matching scores refer to the percentage of the keypoints 
that fall into the accepted regions by the homographies. This figure presents 
the performance of the keypoints on the Bark sequence, which is a series of 
images, rotated and zoomed, of the texture of a tree’s bark.

Occasionally, a small scale space region defined by the criteria of (4.12, 
4.13) may have more than one successful candidate. Although such re-
dundancy can occasionally be useful, it is again biased towards systems 
that produce highly redundant keypoints. To reduce this bias, these 
occurrences are penalised by contributing $1/M$ vote, where $M$ is the 
number of successful candidates within the acceptance region.

The keypoints produced by both product (4.1) and sum (4.2) are fur-
ther evaluated against other keypoint approaches such as Harris Laplace,
Figure 4.9: The Bike sequence contains six images depicting road bikes under different blurring conditions. The images are gradually blurred starting from the first which the finer blur and ending to the sixth which is the coarsest.

Hessian Laplace, MSER and DoG. An approximately equal number of keypoints is sampled ($\approx 1500$) for each detector, to make the comparison fairer. The first comparison is done on the “Bark” Sequence Figure 4.8. The results of this sequence clearly show that product and sum have better repeatabilities than the other detectors, except Harris-Laplace, due to the rotation invariant properties of these keypoints. This sequence contains a bark texture which is successively rotated and rescaled.

The next experiment is the “Bike” Sequence, where a series of images incorporate blur. The first has the least amount of blur, with the sixth image containing the highest amount of blur. It is known that the blur effect leads to global scale change, which is not captured by the homographies used to verify the keypoints. Detectors such as DoG, Harris and Hessian have almost steady behaviour, which a speculation could be that their estimates do not follow the blurring effect which causes global scale change. In addition the sum, product and MSER decay fast, due
Figure 4.10: This figure presents the matching score of the indicated detector by the legend (top right). These images depict a boat rotated and rescaled with the global scale and rotation change are conveyed by the homographies to the fact the artificial scale change being absorbed by the homography. Hence, in the verification of the keypoints, those which truly correspond to the artificial scale change are rejected, producing misleading results in Figure 4.9.

The “Boat” Sequence contains indexed, low quality grey scale images with planar rotation and scale changes which are captured by the homographies. The best performance in this test is obtained for the product (4.1), which is much better from the rest. The second best comes for the sum (4.2) and third best for the Harris-Laplace which, surprisingly, performs better than DoGs. On the contrary, detectors like Hessian-Laplace and MSER perform poorly in this test leading to the fact that its estimates are unreliable, as presented in Figure 4.10.

Next, the “Car” Sequence contains a series of image in gradual decreasing illumination conditions. This test indicates which detectors are invariant to illumination changes. The best performing is the sum (4.2)
Figure 4.11: The Car sequence contains six images depicting cars under different illumination conditions. The images are gradually dimmed starting from the first which the brightest and ending to the sixth, which is the darkest.

producing high repeatability, close to 65%, with second best the product (4.2) and MSER.

The repeatability rates in Figure 4.12 present the performance of the detectors in the Graffiti Sequence. A series of images capture a scene under 3D rotation or a tilt out of plane, co-planar rotation and slight scale changes to fit the scene into the frame. All these transformations are incorporated into the homographies which in turn are used for the keypoint verification. Due to the fact that affine homographies are provided, it is known that the depth component is missing from the homographies matrix and is translated as skew in the affine matrix. This approach misjudges the 3D rotation (slant out of plane) which is not explicitly utilized in the validation procedure. Despite this, the performances of the sum (4.2) and product (4.1) are visibly better than the other detectors, with third best being Harris-Laplace.

Finally, the “House” Sequence involves image compression, where a
Figure 4.12: A scene picturing graffiti under a variety of out-of-plane tilts. This sequence also incorporates scale changes in order to fit on the frame the geometrically transformed scene.

scene of a house is consecutively compressed by the JPEG protocol, increasingly up to 100%. The effect of data compression can be thought as removing structural information from the image. Consequently, this creates plateaus of values or contour value zones which can distort the scale estimation. This sequence evaluates the stability of the scale estimates under poor quality image representations. The sum (4.2) performs the best, with DoG and product (4.1) closely following. The repeatability scores drop quite fast after high amounts of compression. A speculation might be that as the compression increases, the creation of the uniform regions increase, which leads to a perturbation of the scale estimates. In common practice, the edge of the plateau accommodates large scales which may cover half the size of the plateau region. Hence, as the plateaus' size increases, the scale estimates increase, leading to poorer performance.
Figure 4.13: The House scene is a series of compressed images in which each contains an increased compression factor relatively to the previous one. All images are processed with the same compression protocol to produce “jpeg” files.

4.5 Discussion

In this chapter, a method of keypoint detection and scale estimation is suggested and evaluated based on complex filter outputs. The keypoint approach relies on the raw filter magnitudes of each scale, combining the oriented outputs in two alternative ways. The oriented energies of a single scale are summed to produce a scale space where local maxima in \((x, y, \sigma)\) occur. These local maxima are considered repeatable under various scales of the same image allowing the scale invariance to be achieved. A second method, to produce keypoints from such a scale space is also examined by using the product of the oriented magnitude outputs. The localization of these points is performed using a weighted mean of the magnitude outputs of the log-normal filters. This allows sub-pixel and sub-scale estimation, producing reliable scale estimates, as shown in Figure 4.6. Due to frame properties, there is a slight divergence of the scale
estimates near to the borders of the frame. This is corrected by applying consecutive Gaussian 2D functions to tune and refine the filter’s bank scale selectivity. Specifically, the scale error of the estimates is corrected by learning the bias of the filter outputs in which small amounts of random Gaussian noise are introduced.

The repeatability of the detectors was tested under various affine deformations. The keypoint evaluation experiments in the Oxford Affine Detector test data show that the product of orientated magnitudes (4.1) yields the highest repeatability scores, with sum (4.2) being very close in performance. Another interesting property, revealed through these experiments, is that the product has enhanced orientation invariant behaviour comparing to the sum. In particular, the product (4.1) scale space yields higher scores when planar rotation is incorporated in the test. Finally, the repeatability scores are, on average, quite low for all methods, indicating that the scale and orientation invariance of the keypoints has not reached a performance of 90%, concluding that even though maxima detection in scale space is a good approach it may not be sufficient. This leads to the suggestion of other methods capable of achieving higher repeatability scores with an example being the exploration of the underlying mechanisms of saccadic scan-paths and different region focus approaches.

4.6 Conclusions

Overall, this chapter considered how one can achieve keypoint detection derived from Scale Spaces built by log-normal complex quadrature filters. The results shown that repeatability scores outperformed typical
Table 4.2: Summary table of repeatability per detector on the Oxford Affine Test Data. The table presents the repeatabilities’ standard deviation over all affine transformations for each test set. Although the standard deviations are low for the Harris Laplacian detector note that there is a large bias shown in Figures 4.8-4.13.

keypoint detection approaches. This is quite favourable to complex filter approaches which can replace the typical estimation of multiscale gradient fields and their keypoint approaches by more sophisticated methods yielding better performance. This builds up performance in an overall categorization framework where a Gaussian scale space approach is typically favoured as the front end to object recognition.

After the multiscale gradient fields and keypoint detection have been addressed, the next chapter investigates pooling strategies and the construction of descriptors.
Chapter 5

Local Pooling Arrangements

5.1 Introduction

The use of keypoint-triggered patch descriptors has grown into a very successful approach to efficiently represent local visual information for tasks such as image matching [63], [7], image retrieval [63], [7], [87] and object recognition [87], [58]. Descriptors usually represent the orientation of the local gradients in the form of a vector. This chapter investigates the use of complex filter outputs to yield acceptable performance for categorisation tasks. Several pooling arrangements are compared to assess the relative merits of each approach.

The following descriptors are evaluated:

- A Polar arrangement of pooling region based on Complex Gabor outputs (referred to Gabor Mag (D1) and Gabor Phase (D2)).
- An on-line sparsification method which suppress the weak orientations within a complex jet output. Descriptors incorporating this scheme have the prefix Sparse in front of the descriptor’s short name.
- A series of foveated pooling arrangements from Gaussian and log-normal spatial functions (referred to a Gaussian Foveal (D3), Sparse Gaussian Foveal (D4), LogFoveal (D5), Sparse LogFoveal (D6)).
- Optimization of the relative pooling region distances in the Foveated descriptors (D5 and D6).
- Complex filter band optimization through grid based (SIFT-like) descriptors for improving the performance (referred to as Grid-based (D7) and Sparse Grid-based (D8)).
• A baseline performance is established with the SIFT descriptor to assess specific merits of each descriptor type.

The descriptors are benchmarked in two versions each, apart from the main SIFT descriptor. Responses of the Complex filter magnitudes or oriented energies of the filters are pooled for one version. The other descriptor version gathers the information from the phase and magnitude jointly. The rest of this chapter discusses the findings and presents the results along with the evaluation protocols.

5.2 Related Work

Multiresolution histograms are a successful approach to solve the object scale variability by accumulating information into histograms of fixed size. These histogramming approaches provide a method to represent local information in a vectorised from which is much more compact than the original patch size. Despite this, the most important attribute of the following histogramming methods is the quantization effects that histograms’ bins convey. Significant contribution have been made towards the adoption of such methodologies by [45] and [81]. The authors explored different methodologies to extract information from image patches that are encoded by histogramming approaches. The findings of [45] and [81] indicate that histograms of gradient fields can significantly boost recognition performance. This method estimates local gradient orientations within an image patch which have shown good recognition performance. Gradient fields find common ground with neuroscience in which clusters of neurons have been found to respond to spatial gradient information in early biological vision stages [75], [55], [28].

The SIFT approach is an object recognition framework which integrates a keypoint detection strategy and a histogram of oriented gradients into a solid methodology to selectively locate and describe regions as a vectorised representation of important structures within an image [63]. This work [63] employs a Gaussian scale space along with directional derivative operators to obtain histograms of oriented gradients (HoG). Other approaches, which obtain information from different scale space are discussed by [92], [7], [94] and [16].

Further work on pooling strategies for gradient and Gabor-type responses [16] has suggested that foveal arrangements lead to better de-
descriptor performance. The complex filter outputs were evaluated using various spatial pooling configurations [16]. Although the authors [16], [91] have also optimized the parameters, there is a slight difference in the summation areas in this implementation. Specifically, Equation (3.3) also provides a redundancy criterion to optimize the overlap of the summation regions. Applying this approach, the overlap of the regions obtained in this work was reduced, leading to slightly smaller pooling sizes.

5.3 Descriptor Sampling

The following description applies to all descriptors having the “Sparse” prefix in front of their short name (related descriptors are: D1, D2, D4, D6, D8). Once a keypoint is detected, the corresponding scale estimate is used to set a region of log-normal response space from which the descriptor will be constructed. Each pixel within this region receives a weighting approximately set by scaling the pattern according to estimated scale (up to this stage applies to all types of descriptors). In some versions of descriptor (see sparse prefix or D1, D2, D4, D6, D8), a winner-take-all strategy was applied across the 8 orientations of each spatial location. This inhibits weak responses and the strongest response at each keypoint contributing region is made more dominant. This scheme increases the sparsity in the descriptor vectors. Next, each point in \((x, y, \theta)\) is sampled within the summation regions to produce sparse descriptors as discussed in the following sections. Similar to a decision tree, the magnitude votes are distributed by Equation (5.1) where each node corresponds to \((x, y, \theta)\) and subsequent leaves represent phase. Phase is broken down to its dominant projections by Equation (5.1) where these projections are sampled to create a phase descriptor.

\[
\pi(x) = \max_{q=0}^{Q} \left[ \|C(x)\| \cdot |\cos(\angle C(x) \pm \frac{q\pi}{2})| \right]
\]  

The phase of the complex filters plays a role in localizing extrema in image space, where the magnitude of filter responses encodes the dominant orientations along the image plane. Thus, a biological complex cell in mammalian primary visual cortex, which is known to display orientation selectivity and invariance to small position shifts, can be approximated by using magnitude (approximately phase invariant) responses [75],[55],[28].
5.4 Pooling Strategies on Complex Gabor Filter Outputs (D1 & D2)

This section describes the descriptors *Gabor Mag* (D1) and *Gabor Phase* (D2), which are an arrangement of lower dimensionality of SIFT descriptor while employing a sparsification scheme which is described in the following sections. The authors of [16] and [70] have shown in their experiments that polar pooling arrangements lead to better performance over regular grid based pooling. Rather than a square sampling grid, the work in this thesis primarily uses complex Gabor wavelets designed on a polar coordinate system, comprised of 8 sectors and 6 different radii. From each sector, two types of alternative descriptor were constructed: a histogram of phase-invariant orientations, as encoded by Gabor magnitude outputs in the 8 directions, and a histogram of phases.

![Diagram](image)

**Figure 5.1**: *Descriptor Construction*: A circular sampling area is defined by a keypoint’s scale and divided into 8 sectors. Each sector is sampled to produce a histogram of complex (cell’s) orientations (magnitude of complex Gabor filter outputs). The phase of the Gabor outputs can also be embedded in the descriptor as information which describes local structure [43]. The bottom left figure shows how max pooling is performed across orientations in a Gabor jet. On the bottom right, the grouping of position shifts and spatial structure is shown.
5.4.1 64-element Magnitude Descriptor (D1)

A histogram of Complex Oriented Gabor (HCOG) captures geometric information encoded by the orientation selective responses of the Gabor filters within a spatially defined sector. This is extracted from only one of the 5 scales; the scale to be taken is set by criteria, so as to permit an easy comparison with SIFT descriptor performance. Each sector samples from the array of 8 orientated Gabor responses. The strongest response across a set of filters adds a vote to each sector’s HCOG. The 8 sectors result in 8 HCOG’s, where each sector is appended to the descriptor in a clockwise direction. Concatenating these histograms with 8 bins each results in a 64 dimensional descriptor. The result is a descriptor with embedded spatial information along with the directional information of the features.

\[
\begin{bmatrix}
\hat{y} \\
\hat{x}
\end{bmatrix} = r_i \begin{bmatrix}
\sin (k \cdot \tan^{-1}(\frac{x}{y})) \\
\cos (k \cdot \tan^{-1}(\frac{x}{y}))
\end{bmatrix} + \begin{bmatrix} y_0 \\ x_0 \end{bmatrix} \tag{5.2}
\]

\[
\frac{2\pi}{N_s} (i - 1) \leq k \cdot \tan^{-1}(\frac{x}{y}) < \frac{2\pi}{N_s} (i) \tag{5.3}
\]

\[i \in \{1, 2, ..., N_s\}\]

This Equation describes the spatial sampling arrangement used for the descriptor construction. The angle permutations allow the approximation of Gaussian weighting when the displacements of the sampling radii are narrower than the circumference. It also enables oversampling the response near to the center which enhances strongly populated histograms using the data near to the patch centre. Pixel-wise displacements around the circumference are computed by the inverse tangent of \(s = 1\) pixel displacement preference over the radius (\(r\)) of the patch provided by the scale. \(k\) and \(r_i\) are specific values of angle and radius (from the center of each patch) respectively, chosen to harvest the responses of the defined circular patch. Having estimated the angle and radius established by a polar coordinate system, this system is displaced according to a global offset which is the location of a candidate keypoint \((y_0, x_0)\). \(N_s = 8\) sectors are used where the criteria of the bounds of each sector can be found using Equation (5.3).
5.4.2 256-element Phase Descriptor (D2)

The phase response across the Gabor jets is also available; the phase-selective descriptor is sampled by dividing the phase space of response of one Complex Gabor into 4 quadrants. Each quadrant is extended in a fashion that responds to phase projections onto the principal axis of the bottom right of Figure 5.1. Equation (5.4) is applied to assign a vote to the strongest phase projection. The \( j \in \{0, 1, 2, 3\} \) represents integers that change quadrants in the phase space.

\[
P_{h}(i) = \begin{cases} 
\max \{ \cos (\phi + j \pi/2) \}, & \forall (2j - 1)\pi/4 \leq \phi < (2j + 1)\pi/4 \\
0 & \text{otherwise}
\end{cases} \quad (5.4)
\]

The Histogram of Phase Projections (HPP) is constructed, but encoded across four extra elements (channels) of the descriptor vector. These channels gather information based on the principal axis of phase. Specifically, 2 channels encode the positive and negative parts of the cosine response and another 2, the parts of the sinusoidal response, respectively.

![Diagram](image)

**Figure 5.2:** Descriptor illustration: These two versions of descriptors are biologically inspired from (V1) to approximate responses of complex cells (64D) and 4 variants of simple cells (256D) [55],[75].
5.4.3 Foveal Arrangements

This section discusses the pooling arrangements of descriptors (D3, D4, D5 and D6) with specific focus on D5 and D6 which will be used to optimise their pooling arrangement. Once a keypoint is detected at location \((x^{(i)}, y^{(i)})\), the corresponding scale estimate, \(\sigma^{(i)}\) is used to select a region of filter response space, \(g(x, y, k, \sigma)\), from which the descriptor will be constructed. Each location – and therefore, filter output – within this region receives a weighting dependent on its position. The generation of the descriptor entries themselves is performed by using inner products in the spatial domain. This may be expressed by:

\[
H^{(i)}(u) = \left\langle |g(x - x_0^{(i)}, y - y_0, k; \sigma^{(i)})|, \Phi(x - x_0^{(i)}, y - y_0^{(i)}; \sigma^{(i)}; m, n) \right\rangle
\]

where \(u = k + m \cdot N_k + n \cdot M \cdot N_k\), where the \(m = 0, 1, 2, ..., M - 1\) refers to angular sectors of space in an anticlockwise direction from the horizontal, and \(n = 0, ..., N - 1\) refer to radial distances in space from the \(i^{th}\) keypoint. The function \(\Phi(x, y)\), weights the contributions of all direction channels to the entries of the descriptor vector \(H^{(i)}(u)\) according to spatial position:

\[
\Phi(x, y; m, n) = e^{-\alpha \left[ \ln \left( \frac{(x^2 + y^2)^{\frac{1}{2}}}{\sigma} \right) \right]^2 - \beta (\theta - \theta_m)^2}, \quad m=0,1,...,7, n=0,1,2
\]

For \(n = 0\), there is no angular variation, and an offset is introduced into the expression for \(\Phi(x, y; \cdot, 0)\) to avoid \(\ln(0)\). The pattern of weighting produced by \(\phi(x, y)\) is best appreciated by displaying \(\max_{m,n} \phi(x, y; m, n)\), which is shown \((M = 8, N = 3)\) in Figure 5.3(d). This illustrates the spatial localisation provided by the different values of \((m, n)\), which generate “lobes” that clearly define the spatial pooling regions. The layout of these regions was optimised by using an \(L_1\) criterion. Although the authors in [16] also optimized the pooling functions’ parameters, there is a slight difference in the summation areas in this implementation. Specifically, an approach quite similar to the optimisation of Equation (3.3) places a restriction on the overlap of the lobes associated with a single keypoint, leading to slightly smaller pooling sizes than reported in [16] and [91]. The values for \(d_1\) and \(d_2\) are set relative to the estimated scale, and the \(\theta_m\) are spaced \(\pi/4\) apart. The parameters are set to \(\alpha = 4\), and \(\beta = 5.55\). It must be emphasised that these are unrelated to the
directional outputs of the filters: each filter output is sampled by all 17 lobes.

![Figure 5.3](image)

**Figure 5.3:** As shown, 17 spatial pooling kernels with positional shifts are used to define spatial regions for generating descriptor elements by pooling directional filter outputs. The fixation point lies on the center of the union of all spatial kernels. Vertical is y Cartesian axis and horizontally the x axis where these have been converted to polar coordinates to map the pooling kernels. For illustration purposes, all 17 kernels have been overlaid into a single image to illustrate the foveal arrangement and the spatial relationships at each lobe against the others.

Note that Equation (5.5) uses only the magnitude of complex filter outputs $|g(\cdot)|$ in creating the descriptor. Numerous authors have suggested that the responses of one class of cells in the primary visual cortex of mammals [28], can be approximated by using magnitude (approximately phase invariant) responses. Specifically, whilst *simple* cells in the primary visual cortex are typically selective to position within the spatial receptive field and/or spatial phase of the visual pattern, *and* the stimulus orientation, so-called complex cells, which display similar spatial orientation selectivity, tend to have invariance to small positional shifts in the directions indicated in Fig. 5.3, and be phase *insensitive*. It is in this sense that the magnitude of complex filter outputs better mimic biological vision than the linear processes involved in partial derivative
estimation of first-order partial derivatives of Gaussian scale-space.

Equation (5.7) provides a criterion by which the overlap and the spacing of 17 pooling kernels \( \Phi_{1,...,17} \) may be optimised. The optimisation is done at the highest scale size to avoid numerical fluctuations in the small kernels.

\[
L(m, n, \sigma_\rho, \sigma_\theta) = \int_0^1 \int_0^\pi \left( \| \Phi(x, y; m, n) \|^p_p - \mu_{\hat{\Phi}} \right) d\rho d\theta + \lambda \mu_{\hat{\Phi}} \tag{5.7}
\]

Equation (5.8) describes the average frame energy (\( \| \Phi(x, y; m, n) \|^p_p \)) (as introduced in Chapter 3) level of all kernels in order to control the relative basis of overlap among the \( \Phi_{j=1,...,17} \) applied in the spatial domain.

\[
\mu_{\hat{\Phi}} = \int_0^1 \int_0^\pi \| \Phi(x, y; m, n) \|^p_p dx dy \tag{5.8}
\]

The \( p \)-norm, \( p = 2 \) should be taken over the variables \( m \) and \( n \). The final optimized 16 peripheral values are the first radial group of functions (ring) \( d_1 \approx 0.3 \) of patch radius with \( \sigma_\rho \approx 0.35, \sigma_\theta \approx 0.3 \) where the \( \sigma_\rho, \sigma_\theta \) parameters control the spread of the functions in radial and angular fashion respectively. The outer (radial group of functions) with radius \( d_2 \approx 0.7 \) and \( \sigma_\rho \approx 0.35, \sigma_\theta \approx 0.3 \) their corresponding spread. Alternatively, similar spacing of these kernels functions must cross each other at approximately 0.45 normalised to the peak (for validating the optimisation). Finally, after the spacing and overlapping is set, the kernels are renormalised to sum to 1 (\( L_1 \)norm). A unit length radial region described by functions \( \Phi \) are spanned to hold subspaces of a patch as shown in Figure 5.3. The support values of the pooling functions in the center is zero where the max \( x \) and \( y \) are 1 and diagonally \( \sqrt{2} \) in the Figure 5.3 in which 17 lobes representing subspaces of the radial initial field.

In addition, the spanning of the region as defined in Equation (5.6) is optimized using three parameters \((d_1, d_2)\) and overlap of the pooling functions \( \mu_{\hat{\Phi}} \) to maximize the AUC metric per class on the Pascal VOC 2011 dataset. The \( \mu_{\hat{\Phi}} \) parameter can be approximated by Equation (5.9) as the highest common border of these subspaces, if they are normalised to their peak.

\[
s = \max_{x,y} \left( \bigcap_{m,n} \Phi(x, y; m, n) \right) \tag{5.9}
\]
Another two parameters set the distance of the peripheral functions (subspaces) where this can viewed in Figure 5.3 as two sets of angular components ("lobes") grouped into radial sectors. The parameters \( r_1 = d_1 \) and \( r_2 = d_2 \) seek the best spacing from the central lobe. The \( s \) in Equation (5.9) is a scalar value which represents the crossing point of the distributions radially and angularly. This is also referred as overlap because indication at which point the pooling distributions overlap each other.

Unfortunately, the most suited spanning of the lobes was found to be class specific. Thus, the average AUC (Area Under The ROC curve) metric was adopted to indicate the best overall spacing. A similar approach for descriptor optimization have been used by the authors of [100]. One might assume that there is a single global optimum point for a given database. In these experiments, this is not the case because the optimization function \( F_{AUC}(r_1, r_2, s) \) never reaches a optimum point as the parameters vary. Instead, the optimization process exhibits asymptotic behaviour as the second term of 5.10 never reaches an optimum \((r_1, r_2, s)\)
point or \( \varepsilon = 0 \).

\[
|\nabla AUC(r_1, r_2, s)| \leq \varepsilon \tag{5.10}
\]

A solution to this problem is to monitor the gradient ascending values as the \( AUC(r_1, r_2, s) \) outputs higher performance. The optimisation process is stopped if the gradient of Equation (5.10) for an average steep ascent of \( \varepsilon \approx 1\% \) decays to \( \varepsilon \approx 0.1\% \). The start of asymptotic behaviour is highlighted by the blue dashed line in Figure 5.4. The parameter around this border are close to values derived by the optimization on a non-informative field using Equation (5.7). This leads to the conclusion that a generally optimized descriptor can be achieved by simply using Equation (5.7).

### 5.5 Sparsity Inducing Scheme

This section describes the maximal suppression applied to descriptors (D1, D2, D4, D6 and D8) as a technique to sparsify the vectors to a few active units. Two different treatments of complex filter outputs from the tuned filters are used. In the first, raw complex filter outputs are left as is, with no normalisation. In a second version of the descriptor, a winner-take-all (WTA) strategy is applied across the 8 orientations of each spatial location.

\[
\forall (x, y, \sigma_j), \quad k_{\text{max}(x,y,\sigma_j)} = \arg\max_{k \in 0,1,...,7} g(x, y, k, \sigma_j) \tag{5.11}
\]

Then, the response field is modified so that:

\[
\forall (m, n, k), \quad g(x, y, k, \sigma_j) = \begin{cases} 
0 & : k \neq k_{\text{max}} \\
g(x, y, k_{\text{max}}, \sigma_j) & : k = k_{\text{max}}
\end{cases} \tag{5.12}
\]

This completely inhibits the weak responses, so that the strongest response at each position in space becomes the only contributor. This results in a sparser response space, and also sparser descriptor vectors; the average size of these descriptors is 32 elements, indicated in Table 5.4, and also by the keyword “sparse”, used in the experimental results.

\[
|C_{sp}(x, y, k)| = |C(x, y, k)| \cdot \delta_0(k - \arg\max_{k'} C(x, y, k')) \tag{5.13}
\]
Another generalized form of maximal suppression can be seen in Equation (5.13), where a delta function leaves the maximum orientation response active, while suppressing all others. This implementation explores the minimum active orientations with high descriptor performance.

**Figure 5.5:** An illustration of maximal suppression across orientations. The coloured circles indicate patches of different orientations. A sample of eight orientations at a point \((x, y)\) is processed to eliminate all non-maximum orientation responses. This process is performed independently for each scale, rather than across scales.

### 5.6 Descriptor Tuning

The descriptors are evaluated on two databases, the Pascal VOC 2011\(^1\) and the Caltech 101\(^2\). In this evaluation, the basic performance measure is the AUC metric. The Area Under the Curve is computed on a typical Receiver Operator Characteristic ROC curve. The curve is produced by the cumulative distributions of intra-class distance distribution and inter-class distance distribution. The “intra” term refers to those descriptors which belong in the examined category and the “inter” term refers to the descriptors which come from the remainder classes of the dataset. The similarity measure which is used to perform distance distribution is Euclidean distance. The inter-class descriptor distances can be related to the true positive rate of the ROC curve and the inter-class distance distribution can be the false positive rate. The AUC metric is used in this Chapter and elsewhere because it allows comparison with methods in another study [100].

---

\(^1\)http://pascalin.ecs.soton.ac.uk/challenges/VOC/

\(^2\)http://www.vision.caltech.edu/Image_Datasets/Caltech101/
Figure 5.6: (a) An example of good class separation in which the corresponding AUC metric is above the diagonal dashed line. (b) illustrates the opposite effect where the examined class has higher distances distribution yielding a low AUC metric.

5.6.1 Descriptors By Complex Gabor Wavelets

The first comparison is performed on the Pascal VOC 2010 database using the descriptors (D1 & D2) discussed in Sections 5.4.1 and 5.4.2. Complex Gabor wavelets are implemented to generate five scales spaced every octave. Each scale incorporates 8 orientations along with maximal suppression, to produce 64 element descriptors derived from the magnitude response of the complex filters. Another version of the same pooling strategy is the phase and magnitude of the complex filters with a descriptor size of 256 elements. The AUC metric indicates good class separation using descriptor comparisons per class.

Table 5.1 presents a detailed performance comparison per category. Starting from the overall performance, it is very clear that the “Gabor Mag” (D1) descriptor having size of 64 elements performs better than the “SIFT” descriptor with 128 elements (see Table 5.1). The comparison is established by using the generated keypoints of a DoG detector. The radii of the descriptors are all equal for each scale for even fairer comparison. An upside to Gabor Mag (D1) descriptor is that the polar sampling grid fits within the rectangular region of the SIFT descriptor. This adds a further advantage to Gabor Mag (D1), where a smaller region with low
vector dimensionality can produce better class separation results (see Table 5.1).

<table>
<thead>
<tr>
<th></th>
<th>Aeroplane</th>
<th>Bicycle</th>
<th>Bird</th>
<th>Boat</th>
<th>Bottle</th>
<th>Bus</th>
<th>Car</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gabor Mag (64)</td>
<td>0.5692</td>
<td>0.5075</td>
<td>0.4995</td>
<td>0.5374</td>
<td>0.5343</td>
<td>0.5260</td>
<td>0.5068</td>
</tr>
<tr>
<td>Gabor Phase (256)</td>
<td>0.5692</td>
<td>0.4470</td>
<td>0.3869</td>
<td>0.4831</td>
<td>0.4762</td>
<td>0.5012</td>
<td>0.4780</td>
</tr>
<tr>
<td>SIFT (128)</td>
<td>0.5197</td>
<td>0.5101</td>
<td>0.5220</td>
<td>0.5315</td>
<td>0.5231</td>
<td>0.4937</td>
<td>0.5094</td>
</tr>
<tr>
<td></td>
<td>Cat</td>
<td>Chair</td>
<td>Cow</td>
<td>D-Table</td>
<td>Dog</td>
<td>Horse</td>
<td>Motorbike</td>
</tr>
<tr>
<td>Gabor Mag (64)</td>
<td>0.5205</td>
<td>0.5166</td>
<td>0.5397</td>
<td>0.5193</td>
<td>0.5484</td>
<td>0.5174</td>
<td>0.5364</td>
</tr>
<tr>
<td>Gabor Phase (256)</td>
<td>0.4159</td>
<td>0.4600</td>
<td>0.4478</td>
<td>0.4502</td>
<td>0.4121</td>
<td>0.4397</td>
<td>0.4289</td>
</tr>
<tr>
<td>SIFT (128)</td>
<td>0.5402</td>
<td>0.5053</td>
<td>0.5235</td>
<td>0.4947</td>
<td>0.5164</td>
<td>0.5162</td>
<td>0.5124</td>
</tr>
<tr>
<td></td>
<td>Person</td>
<td>P-Plant</td>
<td>Sheep</td>
<td>Sola</td>
<td>Train</td>
<td>Tvmonitor</td>
<td>Average</td>
</tr>
<tr>
<td>Gabor Mag (64)</td>
<td>0.5384</td>
<td>0.5114</td>
<td>0.5425</td>
<td>0.5189</td>
<td>0.5153</td>
<td>0.5362</td>
<td>0.5268</td>
</tr>
<tr>
<td>Gabor Phase (256)</td>
<td>0.4908</td>
<td>0.4364</td>
<td>0.4648</td>
<td>0.4914</td>
<td>0.4763</td>
<td>0.5023</td>
<td>0.4639</td>
</tr>
<tr>
<td>SIFT (128)</td>
<td>0.5115</td>
<td>0.5193</td>
<td>0.5050</td>
<td>0.5182</td>
<td>0.5185</td>
<td>0.4962</td>
<td>0.5143</td>
</tr>
</tbody>
</table>

Table 5.1: Complex Gabor wavelet performance for descriptors (D1 & D2) on Pascal VOC 2010 Train Set compared with SIFT descriptor.

In contrast to magnitude, the phase descriptor Gabor Phase (D2) incorporates more local information than the phase invariant magnitude. Adding the phase to the descriptor increases its size to 256 elements. The average performance of this descriptor is lower than the SIFT descriptor. A possible reason is that phase adds more texture information, increasing the intra-class distances. This may lead to the conclusion that phase-invariance, which is displayed by the magnitude is a necessary adoption to improve the class separability. Overall, it can be safely concluded that using Complex Gabor wavelets without any exhaustive optimization in the frequency domain, can yield good results at a low descriptor dimensionality (64 elements).

5.6.2 Log-normal Outputs vs Gauss-Derivative Outputs

The following experiment adopts SIFT descriptor pooling arrangement as discussed in Chapter 2 where the gradient pooling is replaced by the oriented magnitudes of the log-normal filters (SIFT, D7 & D8 ). Descriptors D7 and D8 adopt only the grid-based pooling of SIFT without a Gaussian weighting and interpolation of the SIFT implementation. Chapter 3 discusses how Complex log-normal responses can be built in Fourier space in order to obtain the desired scale space decomposition that a scalable classification framework should have. It is worth repeating that the log-normal distributions present better coverage characteristics in the frequency domain; this is shown by adopting uniform frame properties (as presented in Chapter 3).
Table 5.2: Band optimization of the log-normal Filters using (SIFT & D7) on the Pascal VOC 2011 Train Set. Band refers to the radial frequency crossing point of the filters as defined in Chapter 3 and illustrated by Figure 3.2.

The previous statement can be verified by a series of experiments from which a conclusion can be drawn by the AUC measures shown in Table 5.2. The experiments are designed to show any difference in terms of classification performance between a Gaussian scale space with directional derivatives and a scale space produced by complex filters in which the directions of derivatives are replaced by filters’ orientations.

The experiment in Table 5.2 is performed using SIFT’s rectangular grid to sample the oriented gradients. The aim of this experiment is not to evaluate descriptor performance based on different arrangements, but to improve the performance of the Complex filters against the Gaussian derivatives using the same descriptors and keypoint locations. Specifically, a DoG detector is applied per image where its outputs \((x, y, \sigma)\) are utilized to harvest the descriptors within the constructed scale spaces.

The main outcome of this comparison is to show whether it is possible to acquire responses that allow better performance over the typical implementation of Gaussian scale space with gradient fields. The same descriptor arrangement with no canonical normalization is deployed to harvest gradient information of identical spatial and scale positions in both approaches. Specifically, half octave spacing is applied to both as scale shifting this sets a solid comparison of which scale extraction leans...
towards better performance. Then, for each scale directional derivatives are applied where their magnitude is divided into 8 orientations. Equally, 8 orientations are adopted for the log-normal responses in half Fourier space. According to the DoG detector predicted scale, the nearest computed scale is sampled using for the SIFT, D7 and D8 descriptors. The sampling process does not consider Gaussian weighting, but the raw outputs of the filter responses. The results on Table 5.2 are conclusive, showing that the bandwidth behaviour is class specific. The best three results of band optimization based on the average AUCs are further tested using the maximal suppression. In addition, the Figure 5.7 shows, using a “jet” colormap to indicate band areas of high class separability. Due to class specific nature of the bandwidth change, the average category performance per redundancy level is employed to infer the best performing overlap of the bands. The best three bandwidth overlaps are further selected to re-evaluate whether there are greater performance gains by applying maximal suppression. Table 5.3 presents the results after the processing step of maximal suppression. The results indicate that, on average, the processing module (complex filters) increases the class separ-
Table 5.3: This table presents the results of log-normal filters after applying the maximal suppression (This is descriptor D8) to the top three best performance bands. Specifically, from the Table 5.2 the top 3 average AUCs are selected and processed again. The maximal suppression treatment boosts the performance, on average.

rability, implying better classification rates. It is noticeable in some cases that bandwidths of lower overlap, leading to sparser responses that can be greatly boosted by the maximal suppression module.

5.7 Descriptor Evaluation

The next test sets a constant bandwidth of filters and deploys foveal arrangements for gradient pooling. The log-normal filter bandwidth is set to the best suited overlap which achieves a uniform frame using \((p = 2)\) Equation (3.3). Next, the log-normal responses are harvested using the product \((4.1)\) for keypoint detection. A log-normal weighting function in foveal arrangement (D5 & D6) as described in the previous section is applied to accumulate the gradient responses into descriptor bins. A similar pooling arrangement is implemented by deploying Gaussian func-
Table 5.4: Descriptor Performance AUC% of \((D3, D4, D5 \text{ and } D6)\) using descriptor distances in Pascal VOC. Two pooling function are used where either Gaussian weights \((D3 \& D4)\) or log-normal weights \((D5 \& D6)\) accumulate gradient responses.

The results in Table 5.4 present the performance of these descriptors where on average, the \(\text{Log Foveal}\) pooling performs better than the Gaussian pooling. This might be due to the tails of the skewed distributions providing better 2D coverage. This claim is also verified when one can achieve uniform frames with low redundancy using uniformity pursuing functions as Equations (3.2) and (3.3). Another significant observation is that the maximal suppression module increases the average performance of descriptors.

After having tuned (the Band of the filter and the descriptor arrangement) on the Pascal VOC database, which only contains 20 classes, a key question is how the descriptors would perform on a larger category test set. Note that although the Caltech 101 database does not contain realistic clutter, the VOC database does, and the performance in the previous section is based on standard train-test paradigm.

The exact experimental (AUC patch measurements related to classes)
set up is applied to the Caltech 101 database to assess descriptor performance on a more diverse environment which contains 101 categories. The two pooling functions Gaussian and log-normal are adopted again. Due to the large number of categories the AUC output of each method is sorted per class. This helps to visually compare the performance of each method, as illustrated in Figure 5.8. The SIFT’s AUC curve is taken as a baseline performance, where the other descriptors perform poorly for some categories. After 40 categories, both descriptors surpass the AUC curve of SIFT. The AUC curves of Sparse LogFoveal and Sparse Gaussian reach and outperform SIFT’s performance even at the 10th category. This is a very encouraging indication that maximal suppression indeed increases descriptor performance.

Overall, the Log-Foveal descriptor performs better than Gaussian functions and outperforms SIFT descriptors. Also, the maximal suppression module increases the overall performance when deployed in the sampling
process, but also produces sparse descriptors where only 32 non-zero elements are active. Thus, sparse vector and matrix libraries can be utilized to speed up further descriptor processing.

5.8 Discussion

Local pooling strategies attempt to seek effective pooling spatial arrangements of gradient fields. Some of these strategies are tested in this chapter on complex filter outputs. The experiments addressed whether such responses perform better than a typical implementation of a Gaussian scale space with directional derivatives. The experiment in Table 5.1 showed that complex Gabor filters perform better than a typical SIFT implementation using common DoG detector and equivalent descriptor size variation. Although these filters are not optimized for classification performance, they marginally outperform the SIFT descriptor having lower dimensionality than SIFT.

The experiment in Figure 5.4 involves the optimisation of the LogFoveal arrangements. This experiment attempted to find the best spacing of the pooling regions which are deployed by the LogFoveal arrangements. It was found that this exhaustive optimisation exhibited class specific nature. Thus, the average performance from all classes was employed to yield a generalised arrangement of the LogFoveal arrangements. The optimised form yields marginal improvements in comparison with the proposed optimisation in this Chapter.

The experiments, in Figure 5.7 and Tables 5.2 and 5.3 are performed by harvesting the magnitude outputs of complex log-normal responses with different amounts of band overlap. This was done by restricting the radial and angular bandwidth of the filter to the indicated redundancy level in Table 5.2 for each test. These responses are harvested by rectangular SIFT descriptors at the exact scale spacing and comparable descriptor size per scale. The results show that log-normal scale space outperforms the Gaussian approach with directional derivatives. Also, this test reveals that the level of redundancy (overlap of distributions) exhibits a class specific nature. Hence, average performance is employed to address the scale space performance. Next, the best three redundancy levels are selected to further test the effect of maximal suppression module. It is surprising, once more, that this approach boosts
the performance even more where the relative scale space performance is broadened to approximately 5%.

Finally, foveal arrangements using either log-normal or Gaussian weighting functions were applied to the log-normal scale space. In these experiments (see Table 5.4 and Figure 5.8), yielding the descriptors LogFoveal (D5 & D6) and Gaussian (D3 & D4) LogFoveal descriptors performed better than Gaussian spatial weighting in many cases. Also, the maximal suppression again improved the descriptor performance over raw outputs. The same methodology was repeated on Caltech 101, where LogFoveal descriptors improve marginally over Gaussian. The improvement was consistent in both databases. In addition, the descriptor performance was further improved when the maximal suppression was incorporated into the descriptor construction. Overall, log-normal space behaves better than Gaussian scale space in the frequency and spatial domain becoming an attractive approach to descriptor sampling methodology as well.

| Table 5.5: This table presents the summaries of the different descriptor arrangements tested on Caltech 101 and Pascal VOC 2011. It is notable that the best performing arrangement is the LogFoveal. Filter bank optimization can give further improvements to overall performance. |
|---------------------------------|-----------------|-----------------|
| Gabor Mag 64 (D1) | 0.5268 | Gabor Phase 256 (D2) | 0.4639 | LogFoveal Mag 136 (D5) | 0.5449 |
| Gaussian Mag 136 (D3) | 0.5428 | Sparse Gaussian Mag 136 (D4) | 0.5624 | Sparse LogFoveal Mag 136 (D6) | 0.5593 |
| Grid-Based 128 (D7) | 0.5385 | Sparse Grid-Based 128 (D8) | 0.5750 | SIFT 128 | 0.5173 |

5.9 Conclusions

The experiments suggest that redundancy affects the descriptor performance of the foveated pooling functions. Maximal suppression always yielded improvement over raw filter outputs for any spatial pooling technique and weighting function. LogFoveal descriptors showed better average performance than other approaches. The pooling method provides a means of capturing valuable information about patch structure that can be converted into a feature vector. The feature vector, like a SIFT descriptor, is a very compact representation which makes possible the
adoption of machine learning approaches for further processing. Especially, this is true when large databases can produce millions of descriptor which their massive number discourages look up over their number. In this case, a clustering approach vector quantizes the descriptors into “mean” vectors representing cluster centres.
Chapter 6

Vector Quantization -
Clustering

6.1 Introduction

Clustering is an unsupervised learning method which enables us to assign observations of descriptors to a group (cluster) of descriptors in a training sample. Statistical analysis of descriptors, as well as comparisons of query and database descriptors, is very difficult to perform efficiently due to the high-dimensional nature of most image patch descriptors and the large number of observations; consequently, clustering algorithms are usually applied to image descriptors to group representative subsets of observations in descriptor space. This allows faster comparison to be done between within-class and between-class image patches, and also to estimate the density of descriptor space and how it changes between different patches and objects. One can view the clustering as a vector quantisation, and also as a form of descriptor compression method. Grouping of descriptors also allows efficient retrieval mechanisms, for example sample by applying ranked retrieval techniques developed for fast indexing [87], [73].

6.2 Related Work

There are numerous algorithms that attempt to find the optimal number of clusters. One common approach arising from graph theory is spectral clustering. Assuming that samples are connected under some pre-defined relationships, a connectivity matrix can be obtained describing those re-
relationships. Usually, the eigenvectors of this matrix are further grouped into clusters according to their magnitude. Then the eigenvectors correspond to subgraphs with similar connectivity [72], [5].

Other methods involve a probabilistic approach to clustering, in which prior assumptions of Gaussian distributions in the descriptor feature space are employed to discover the best number and the parameter of mixture components which best represent the database [77]. Also, a closely related approach is the Mean-Shift algorithm in which Gaussian density kernel is employed to seek modes in feature space [22]. Typically, each mode corresponds to a cluster centre where the cluster’s width is defined by the samples which comprise the kernel.

A sparse coding approach [79] has also been used to derive sparse codes that can represent vectors with the smallest numbers of active units. Usually, minimization of reconstruction error is sought in conjunction with regularization of the sparse codes [79]. Also, the minimization of the reconstruction error has been explored by vector quantization approaches. These usually overlap with $k$-means approaches, which attempt to cluster samples based on a nearest neighbour approach. A typical implementation of $k$-means would involve the use of Euclidean distance between the samples, which in turn are assigned to clusters. J. Sivic and A. Zisserman report that $k$-means is very reliable [88], although in some cases the resulting equi-sized clusters are undesirable. A problem with many clustering approaches is the optimal choice of number of the clusters, although methods have been proposed to estimate this [76]. Due to the fact that the previous authors have reported good cluster quality with $k$-means whether the number of data needs to be reduced, D. Nister and H. Stewenius [73] showed that this can be achieved in a structured manner. A typical problem of such implementations is the computation time, for which the authors of [83] proposed a suitably approximate solution, speeding up the $k$-means algorithm.

In our implementation, descriptors represent image patch gradient observations or orientations. If we were to harvest 1 million descriptors from an image database, it would be difficult to perform further analysis. Thus, descriptors are compressed to visual words by clustering. The clustering algorithm raises issues as to the representability of the descriptor database due to the quantization of descriptors to visual words.
6.3 Histograms of Visual Words

In image retrieval, vast numbers of image vectors are used to assign cluster IDs, which are then used to perform fast indexing [88],[73]. This enables fast search within an image database, by significantly reducing the time spent to search the data. Furthermore, the quantised vectors allow the application of approaches [76],[88] such as text document retrieval, whereby images are treated as documents, and the frequency of occurrences of each cluster plays a part in assessing the relevance of results to a query.

Histograms of visual words can be created to estimate the word frequencies from each category of patch that might be found in an image or video sequence [89]. The term Visual Words refers to quantised vectors that describe various clusters which can be found throughout the database. Estimating discrete density functions (histograms) within each image enables one to categorize the image as an observed vector, minimizing the look up time. Many of these words may appear often in the database across categories; instead of using stop lists [87], we use histograms over visual words weighted according to Term-Frequency, Inverse Document Frequency (TF-IDF). This well-known weighting, is defined by:

$$t_i = \frac{n_{id}}{n_d} \log \frac{N}{n_i}$$  \hspace{1cm} (6.1)

The term \(n_{id}\) represents the number of occurrences of word \(i\) found in an image. This is divided by the total number of the words, \(n_d\), within the examined image. The logarithm of the number of words in the database, \(N\), over the measured occurrences of that word, \(n_i\), is used to weight each word in the 500 dimensional vector encoding of each image (for all experiments in this Chapter).

6.4 An Evaluation of Cluster Validity and Structure

Two separate experiments designed to assess the inter-cluster distance distribution vs the intra-cluster distance distribution as an internal cluster evaluation, using the 2010 standard Pascal VOC database. Clustering analysis methodologies have proposed which all have common ground of intra-inter cluster distance distribution. These distances are calculated
using various measures. A more simplified methodology is adopted to reveal basic cluster attributes by employing intra cluster distance. The term “intra” refers to a characteristic that lies within each cluster which in this case is measured by Euclidean distance. The distance of each cluster centre is measured against all the cluster members that comprise the examined cluster. This produces an average distance from the centre, which is related to cluster spread. The cluster’s variance of the distances indicates the amount of concentration of the cluster members. Specifically, high variance of the distances indicates scattering of the cluster members and low variance show densely populated cluster. Another view of variance is the information of the size of the cluster even though this can be estimate by taking into account its members.

The term “inter” refers to the characteristic which is found between clusters. In this case, the Euclidean distance is applied to the cluster members to reveal the spread of the clusters across the feature space. The cluster centre distance variance will indicate the scattering of the clusters. For consistency, the inter-cluster distances are computed using the 10 nearest neighbours of an examined cluster centre. Otherwise, the inter distance will be a global prediction which will not provide information of the distance or overlap of adjacent clusters.

Finally, principal component analysis will be applied to the cluster centres to further illustrate and graphically explain the findings on “intra” and “inter” cluster distance distributions. The principal component analysis serves as a dimensionality reduction method; due to the high dimensionality of the feature vectors, it is impossible to visualize their relationships in feature space. PCA (Principal Component Analysis) can be used to identify the most variant dimensions among the cluster centres, which will be illustrated in the second set of experiments. The top two variance components are acquired through eigenvalue decomposition of the covariance matrix formed from the cluster centres. This approach allows one to visualise a projection of the feature space along the direction maximum scattering found in the feature space.
Figure 6.1: An example of annotated images in VOC Database along with their bounding boxes per object class.

The Pascal 2010 database\(^1\) contains 11300 labelled images in twenty classes across multiple poses. In order to ensure that comparable conditions were used to cluster descriptors, an equal number of descriptors per class is accumulated into a stack (300K approx). The inter and intra cluster distances are calculated to identify the effect of different scale spaces and descriptor sampling techniques in their feature space.

\(^1\)http://pascallin.ecs.soton.ac.uk/challenges/VOC/
Figure 6.2: This is the distance distribution of the SIFT descriptor with no angular shifting of the descriptor. The broad coloured area along the curves indicates the deviation of distance per cluster centre (visual word).

The first experiment uses SIFT descriptors with no rotation invariance. Typically, the rotation invariance in a descriptor does not enhance the overall classification performance, at least in the VOC context. Specifically, Figure 6.2 shows the internal evaluation of the clustering tendency of the descriptors using $k$-means to derive the clusters. The descriptor in this example does not incorporate rotation invariance at the descriptor level. It is clearly visible that the distance within the clusters (intra-cluster, blue) is lower than the distance between the cluster centres (inter-cluster, red). This is reasonable, as shown by the intra-cluster distance in which the tendency of the curve is steady between 50-400 visual words. This translates (steady variance and distance) to equisized clusters which is the objective of standard $k$-means implementations. In addition, the inter-cluster distances show good separation among the clusters with the deviation indicating the degree of overlap (red area).
Figure 6.3: A dictionary of Complex log-normal wavelets with optimised bandwidth produces its scale space which, in turn, is applied in the classification framework. The descriptors are grouped into 500 clusters to produce the result shown in this figure.

The next experiment, as shown in Figure 6.3, involves SIFT descriptors (grid-based version) on an optimized log-normal scale space. These are imported to the $k$-means algorithm to produce 500 cluster centres. The intra-cluster distance is equal to all clusters, which means that equipisized clusters have not been obtained as should be by $k$-means. A similar trend is shown by the inter-cluster distances in which both distances indicate that this feature space does not extend in a linear fashion. Despite this, scattering behaviour in feature space leads to better class separability.
The SIFT descriptor includes rotation invariance. The same number of clusters (500) organise the descriptors into groups.

The SIFT descriptors are re-clustered by the $k$-means algorithm where, in this case, rotation invariance is included in the descriptor structure. The results in Figure 6.4 indicate that this modification leads to a uniform feature space with equisized clusters. This conclusion can be drawn by observing the distributions that present steady behaviour (steady distance outputs in the examined Figure). Also, there is no major overlap among the clusters, which is indicative of the feature space being well segmented. Although the good separation of inter and intra cluster distance distributions the class separability in practice is rather lower than the rotation selective version (as shown in the previous chapter and will be shown in the next chapter too).
Figure 6.5: This figure illustrates the cluster separation using Log-Foveal descriptors on the produced log-normal nonlinear scale space.

Figure 6.5 presents the cluster distance distributions using Log-foveal descriptors. From this result, it is noticeable that the inter-cluster distances are close to the maximum possible. This indicates a high spread of the clusters which, in turn, occupies a larger part of the feature space. The intra-class distances also yield high values which translate into large clusters. A new property of the clusters, produced with this option, is their increased size, which should be noted. These descriptors yield the second highest class separability, highlighting that scale space as important factor of the performance.
Finally, the maximal suppression is embedded into Log-Foveal descriptors, where a million descriptor samples are clustered into 500 clusters. Both inter and intra cluster distance distributions show similar behaviour. The intra-cluster distance decreases, indicating that the size of the clusters changes. In addition, the inter-cluster distance decays as well where the cluster concentration changes. Actually, from both curves, it is very clear that a Voronoi tessellation (clusters of equal size) applied to this feature space will not be adequate to properly capture representative groups over this feature space. Even though the feature space varies with the samples, these descriptors have yielded the best performance on class separability tasks (shown in the next chapter).
In the following experiments, PCA is applied on the cluster centres as a dimensionality reduction approach in order to further explore and validate the findings of cluster distance. The two components of the highest variance are plotted in each example, to investigate the effect of cluster scatter on a features’ space projection. For illustration purposes, a fixed cluster size has been adopted. The cluster radius is the average distance between adjacent clusters in the 2D space obtained by the two components of the PCA.

The Figure 6.7 illustrates a projection of feature space by using SIFT descriptors with no rotation invariance. In conjunction with the findings of Figure 6.2, the visualisation of the two highest variance components suggests that the cluster distance curves (intra-inter distances) with high separation indicate high concentration of clusters is high in specific feature space regions. The clusters lying away from the centre of the clusters’ mass are thought to correspond to cluster with large variance or many
Figure 6.8: This plot shows the cluster scattering of the SIFT-like (rectangular grid (D7)) descriptors sampled on a log-normal scale space with optimised bands to yield the best class separability.

The Figure 6.8 presents a feature space projection of the rectangular grid sampling on the log-normal scale space. There is a densely populated area in the middle of the illustration where the clusters overlap by large amounts. This can be related to the cluster centres which show low cluster sizes and low cluster adjacency distances. Consequently, it is further assumed that larger cluster sizes and adjacency distances correspond to clusters lying further away from the area of their large concentration.
Figure 6.9: *SIFT descriptors are rotation invariant in this example where the cluster of these descriptors are projected above. It is noticeable how the orientation invariance has changed the cluster distribution.*

Figure 6.9 illustrates the scattering of SIFT descriptors. This example should be compared with that of Figure’s 6.4, where rotation invariant descriptors are clustered. The rotation invariance yields clusters uniformly distributed over the derived feature space. As illustrated in Figure 6.9, the clusters are spread over the projection of the feature space. This is in agreement with the uniform distribution of inter-cluster distances observed in Figure 6.4, which is steady and has relatively large distance values. In addition, the intra-cluster distances have also large values indicating larger clusters than the previous examples. This is illustrated as well in Figure 6.9 where the cluster representations occupy larger regions.
Figure 6.10: In this example, Log-Foveal descriptors sampled by the raw outputs of the log-normal responses are further clustered. Such descriptor clusters are illustrated above.

In Figure 6.10 the clusters for the Log-Foveal descriptors are visualised. These are harvested by the log-normal scale space using keypoints which are discussed in the previous chapter. This cluster representation is related with Figure 6.5 which shows cluster distances. Due to the fact that these descriptors are rotation tolerant than a fixed rectangular grid descriptor (the large sampling region of the polar descriptor exhibits better tolerance to angular shifts compared to a grid-based (D7)), similar cluster distribution in the feature space is present at Figure 6.9. Also, in this example the clusters’ size is large, with the clusters better occupying the feature space. The clusters’ distribution over the feature space is almost uniform, but there is a visible area where the clusters are highly concentrated which is spread horizontally in the middle of the Figure 6.10.
Figure 6.11: The maximal suppression is introduced in the log-normal outputs which are sampled by the Log-Foveal descriptors. The cluster distribution is much different from the previous examples where here the cluster centres are co-organised into elongated lobes.

Finally, Figure 6.11 presents the clusters’ distributions in the projected feature space. The maximal suppression is introduced in the Log-Foveal descriptors, yielding sparse vectorised responses. Their clustering exhibits a notably different grouping behaviour comparing to the previous examples. As the curves show in Figure 6.6, there is always a spread of inter-intra cluster distances implying that there are regions with high and low cluster concentration. Observing the representation in Figure 6.11, it is clear that the descriptors exhibit a tendency of self-clustering in a different and non-uniform way. An explanation of this effect is that maximal suppression forces the scale space the generate responses to a unique orientation for each spatial point.

6.5 Cluster Representatives in the Spatial Domain

There are unique properties of complex log-normal filters which may lead to different performance in scalable, keypoint type approaches. Specif-
ically, complex filters encode four main translations of the filter carrier comparing to the majority of single phase filters which are selective to spatial translation. These translations correspond to angle offsets which are set into the carriers (\(\cos()\) or \(\sin()\)). These functions can transform one into other by creating a cyclic signal outputs in Cartesian coordinates. The quadrature filters contain both components (\(\cos()\) and \(\sin()\)) which enables them to encode all resolvents of \(\cos()\) and \(\sin()\) comparing to derivative filters which usually encode only one of \(\cos()\) or \(\sin()\). An easy way to extract invariant behaviour from such complex filters is through their magnitude.

In the experiments, shown in Figure 6.12, magnitude responses of the log-normal filters are produced to introduce spatial phase invariance (or invariant positional translations) into the categorisation pipeline. The clustering approach is applied on descriptors, obtained by such responses, to group similar properties into clusters. An interpretation can be made about the size of the clusters, which may be related to the variance of these responses. Each cluster is assumed to incorporate phase invariance, coming from the complex filters and structure deformations or occlusions as well. The structural tolerance is cluster specific, which means that a big cluster can tolerate structural occlusions and deformations embedded by the descriptor construction. Thus, complex filters do indeed allow some spatial phase invariance and their descriptors embody local structural information.

The dictionaries are assigned to the descriptors in each image (in the Pascal VOC 2011). The assigned clusters to descriptors are traced backwards down to the image patch to illustrate examples of the grouping effect. Specifically, the visual words (cluster centres) are mapped onto image patches (using the corresponded descriptors) to identify cluster member along with intuitive representations of the cluster centres. The following procedure is applied to both scale spaces log-normal and Laplacian of Gaussian. Each cluster’s members are identified and pooled into a module which computes the correlation among the patches. A correlation confusion matrix is produced which is further grouped into four ranges which within those classified correlation values representative patches are selected to represent four typical examples of the cluster members. In Figure 6.12, it is obvious that the clusters incorporate phase invariance, which also includes invariance to contrast inversion. An important point
Figure 6.12: Visual Words reverse engineered from complex log-normal magnitude to original patches. The patches have been acquired from the Pascal VOC 2011 database.

is that this invariance is grouped into more complex structures as shown in the Figure 6.12. Also, note that curvature, corners, T-junctions and higher order of junctions are embody contrast inversion. Also, there is an example shifting shadowing (4th row) which also incorporated into one cluster. Another property which is due to descriptor clustering in this case is the structural deformation tolerance and occasionally occlusion which is incorporated by the clusters. This observation is especially true for more complex structures such as T-junctions.
Figure 6.13: Visual Words reverse engineered from (SIFT’s) Gaussian scale space with two directional derivatives to original patches. The patches have been acquired from the Pascal VOC 2011 database.

On the contrary, the gradient fields of Gaussian scale space exhibits poorer invariant properties at the clustering level. As illustrated in Figure 6.13, invariance to contrast inversion (spatial phase invariance) is almost absent where contrast sensitivity ranges are grouped in the cluster level. This is consistent with a gradient, scale space not incorporating phase invariance. In addition, it is clear in some examples that the same clustering method and parametrization inherits primary level properties such as illumination invariance. This invariance arise from the filtering level for the complex log-normal filters. Also, Figure 4.11 shows that
Log-normal scale space achieves very high performance in different illumination conditions comparing to Gaussian scale space. In addition, the examples of Figure 6.13 indicates poor structural information conveyance at the descriptor level which consequently leads to decreased structural tolerance by the clusters.

6.6 Discussion

Clustering approaches, and specifically vector quantization, attempts to find representative clusters or quantized vectors (visual words or cluster centres) among the original observations. The clustering approach used in this chapter is $k$-means with a fixed cluster number of 500 clusters. The choice of 500 visual words serves as a representative small size codebook for the previous graphical illustrations any larger codebook size would be inappropriate for illustration. Also the small size code enhances grouping properties which in turn assists better illustrations to the reader. Large codebooks usually develop tolerance to small changes in the patch level becoming difficult to set up an illustration for such codebook sizes. The first test addresses the intra-inter cluster distance distributions where it was found that equisized clusters did not always occur. It is found that descriptors producing high class discrimination, form a non-uniform tessellation in feature’s space (see Chapter 5 Tables). For instance, away from the dense cluster “populated regions”, the distribution of the clusters decreases. This is an important indication that other tessellation methods (typically clustering) using different grouping criteria, e.g. by using different distribution criteria over the feature space may improve the clustering performance. The implication of such findings raises the possibility of applying different feature space partitioning approaches to grouping functions such as the Mahalanobis distance, which considers the variance of the samples. Another approach to tackle this may be high dimensional Gaussian processes which also consider variance, but always use Gaussian priors. Based on the cluster observations visualised by the PCA approach, a log-polar tessellation could be adopted which considers variable partitioning of the feature space but thought must be given on the choice of the similarity measures as well. Another interesting property is that maximal suppression causes the descriptors to group themselves into elongated distributions (see Figure 6.11). Even
though this behaviour is unusual in feature space, it produces high class separability. Such distributions in the feature space can be utilized by clustering methods which do not consider equivariant feature space.

Finally, a more interpretable test is established to inspect typical cluster centres and their most diverse cluster members. These are visually illustrated to highlight any profound differences. Comparing the two (Figures 6.12 and 6.13), the effect of the phase invariant nature of complex filtering can be seen at the visual word level. This property is found among the members of all clusters such as bars, curvature, T-junctions, etc. Also, the clusters exhibit a pseudo structural invariance in both scale spaces. This is more profound for the Complex log-normal scale space, where T-junctions may have occluded parts. On the contrary, clusters of the Gaussian scale space seem to convey illumination invariance properties at this processing level.

6.7 Conclusions

The chapter explores the clusters’ internal properties that lead to improved categorization performance. K-means is the only algorithm which was implemented to perform clustering on the descriptors. Basic properties are attempted to be recovered on the generated clusters which will lead to useful conclusions. The first experiment retrieves the distances within and among the clusters as an attempt to investigate the clusters’ variation across the feature space. The results of this experiment were followed by PCA as a dimensionality reduction in order to visualise the visual words in their feature space. This will assist the visual validation of specific hypothesis upon the distance distributions. Finally, the derived clusters were mapped to the corresponding image patches to visually inspect the cluster members. Throughout these experiments, the descriptors that enhance the class separability are distributed in a non-uniform fashion. These findings lead to speculation on whether other clustering approaches might improve the overall performance of categorisation tasks.
Chapter 7

Incorporating Spatial Information

7.1 Introduction

Histograms of Visual Words (HoVW) have shown reasonable performance in object recognition tasks. Typically, a HoVW describes visual word occurrences in an input or query image. At this stage of processing, quantized representations of a large database are deployed to balance fast indexing, as a look up process, with recognition performance. Lately, research on bag-of-features approaches has shown that incorporating spatial relationships of such features yields noticeable performance gains over simple HoVW techniques. A successful approach to include spatial information is spatial pyramids, which are comprised of spatially restricted HoVWs. This creates the effect of acquiring the HoVWs at different resolution levels and regions of focus. Despite this, spatial pyramids embody coarse spatial relationships. This chapter explores a different perspective for integrating pairwise spatial relationships with visual words.

7.2 Related Work

Bag-of-visual-words approaches represent a methodology which generally improves the overall classification performance. Typically, such approaches deploy the HoVW methodology to provide histograms which contain information about the visual word occurrences of a query image. Such implementations have been adopted by J.Zhang, et al.[102], Sivic, et al.[86] and [67] who explored different scenarios to acquire visual word
information while improving the overall categorisation performance. Li Fei-fei [36] have shown that when the intra-class variability of a database is low, dense grid approaches can yield better performance rates than key-point approaches. Although this is true for such databases, the massive processing that would be needed for classification tasks would be expensive to compute for databases sizes above a million images. A further improvement on bag of visual word approaches introduced by T. Hofmann et al.[48], who applied PLSA, a latent vote probabilistic approach based on prior knowledge of topics in a database. Another probabilistic approach was introduced by Sivic et al.[87], who implements the inverse document frequency term. This voting scheme downweights frequently occurring words which are considered to provide low class discrimination. Further improvements on the bag of word approaches are provided by spatial pyramids which incorporate coarse spatial information in HoVWs concatenations. Grauman, et al.[44] implemented pyramids in feature space which did not improve the overall performance much. Despite this, spatial pyramids were implemented by Lazebnik, et al.[58] who showed much improvement in recognition rates. Finally, Bosch, et al.[12] improved further the classification rates by using HoGs (Histogram of Gradients) of many orientations. Although the previous is unrelated to this [58], the same spatial multilevel organisation was used (spatial pyramids).

7.3 Learning Spatial Pyramids

Spatial Pyramids are a recently introduced methodology by Lazebnik, et al.[58] to subdivide and “disorder” an image into a vectorised representation. This approach converts spatial relationships of the assigned visual words within an image into a vector form. In many reported systems for object recognition, the spatial pyramid implementation has three levels of resolution. The first level encodes a coarse view of visual words coming from the whole image, similar to a HoVW. The second level of the pyramid divides the image into four quadrants. This increases the spatial resolution of the visual words comparing to the previous coarser level. Actually, the HoVW becomes “localized” into the image regions rather than the whole image. The third level of the pyramid is a subdivided version of the previous layer, which contains 16 regions. The third
layer conveys the highest spatial resolution of a candidate image with HoVWs conveying finer visual word information from all the previous levels. Finally, the HoVWs from of all the levels of the pyramid are concatenated into a large vector. A graphic representation of a three level spatial pyramid is illustrated in Figure 7.1.

Figure 7.1: This is a graphic representation of how the image is subdivided into regions. Three levels of a spatial pyramid are presented with increasing resolution of each level of the pyramid.

Each of the images are now represented as large vectors (spatial pyramid representation); typically a machine learning approach is applied to predict the involved classes within each query image. Kernel machines represent a successful approach in machine learning to map the feature space of the vectors into an abstract space where a classifier can learn the separability of the annotated classes. The first step in this learning approach is to reduce the vectors to simple point-pairwise relationships within a kernel. This is one of the main reasons why kernels are so commonly used in learning. Usually, to perform supervised learning one needs an annotated training set, where the training kernel can be constructed. The main property of the kernel which leads to the rules of its construction is its symmetry [84]. Its structure is similar to a Gram matrix. For instance let \( X = \{x_1, \ldots, x_N\} \) be a set of vectors \( x_i \) then the kernel matrix is:

\[
K_{ij} = \langle x_i, x_j \rangle
\]

with \( i \) be row indices and \( j \) be the column indices. The \( x \) is reduced to a point in order to be mapped into the kernel. The mapping of the kernel matrix \( K \) is provided by the similarity functions such as Euclidean distance, Chi-squared distance, Helinger distance, etc, [84]. Thus, the mapping from the original feature space of a vector \( x \) to an abstract
space in the Kernel matrix is defined as $\phi(x)$. Hence, the explicit kernel form of a linear kernel is:

$$x \mapsto \phi(x) = K(x, \cdot) = \left\langle \frac{x}{\|x\|}, \cdot \right\rangle \quad (7.2)$$

Practically, the linear kernel is expressed in terms of inner products of the normalised ($L_2$ norm) involved vector set, which in this case, is the training set. For example, Equation (7.2) may be replaced by a Gaussian Kernel mapping $\phi(x) = K(x, \cdot) = \exp\left(-\frac{\|x-x\|^2}{2\sigma^2}\right)$ which is also widely used. Similarly, the testing kernel is constructed by calculating the pairwise relationships of the training set vectors with the test set with one dimension of the matrix tied to training set and the other dimension to the testing set [84].

**Figure 7.2:** This is a geometric interpretation of how the SVM classifier separates two classes in a binary problem. The coloured dots represent instances of two classes in the feature space. The red ones correspond to negative labeled classes and the blue ones to positive labeled classes. The separation margin between the classes is indicated by an orange line along with its width $\frac{2}{\|w\|}$. The borders of the margin are described by the illustrated equation which a compact form is presented by the Equation (7.3). The purple dashed lines correspond to $\xi_i$, introduced to handle classification noise along the borders. The coloured dots which are crossed by the black lines are called support vector as they are the instances which support the margin. Altered from [26]

Once the kernels have been obtained, a classifier can be applied to the training kernel to learn separate the classes. A commonly adopted classifier for such a classification task is the Support Vector Machine (SVM),
which is very efficient in terms of classification performance. Due to the fact that the SVM is a binary classifier, meaning that it learns to separate the target class from the non-class instances, the multiclass learning is set up by one against the rest of the classes for each category. Specifically, an SVM classifier for each category is trained, where the multiclass evaluation is performed by assigning prediction probabilities from all the trained models (SVMs). The learning process of an SVM is performed on the kernel matrix by adjusting and maximising the width of a linear margin between the target classes and the non-class examples. The core of the SVM typically uses quadratic programming as an optimization approach for maximising a soft margin between the target class and the non-class populations. The algorithm itself utilizes all the instances of the training examples along with their labels (also known as supervised learning when labels involved). The primal form of the margin or hyperplane is expressed by the following equation.

\[ y_i (w \cdot x_i - b) \geq 1 - \xi_i, \quad \forall \, 1 \leq i \leq n \quad \text{and} \quad \xi_i \geq 0 \quad (7.3) \]

The width of the bounded space that is yielded by Equation (7.3) is expressed by \( \frac{2}{||w||} \). Specifically the labels \( y_i \in \{1, -1\} \) define two groups of data where the two borders of the dot product between the norm vector \( w \) and samples \( x_i \) form the slope of the hyperplane while is biased by \( b \) (an absolute offset in the feature space) [26]. Good class separation is achieved by maximising this term \( \frac{2}{||w||} \) which is the width of the margin. Dealing with it as an optimization problem, Lagrangian multipliers are introduced to yield the following expressions:

\[
\min_{w, \xi_i} \left\{ \frac{1}{2} ||w||^2 + C \sum_{i=1}^{n} \xi_i \right\} \quad (7.4)
\]

\[
\min_{w, \xi_i, b} \max_{\alpha_i, \beta_i \geq 0} \left\{ \frac{1}{2} ||w||^2 + C \sum_{i=1}^{n} \xi_i \ldots \right\}
\]

\[
- \sum_{i=1}^{n} \alpha_i [y_i (w \cdot x_i - b) - 1 - \xi_i] - \sum_{i=1}^{n} \beta_i \xi_i \right\} \quad (7.5)
\]

\[ s.t. \quad w = \sum_{i=1}^{n} \alpha_i y_i x_i \quad (7.6) \]
The minimisation of Equation (7.4) actually maximizes the separation border constrained by $C$ and $\xi$. $C$ represents the cost of the border minimization in which high values (with the constraints of $\alpha_i$) of $C$ usually improve thin class borders [26]. $\xi_i$ guarantees the positivity of Equation (7.3) but also handles classification noise on the borders defined by (7.3). Equation (7.5) describes the minimisation of norm $w$ with the data $x_i$ [26]. Its geometric interpretation is the widest possible soft margin between the two classes. The maximization term fits the two borders of the soft margin over the data. Hence, Equation (7.3) can be expanded into two lines which define the borders of the soft margin ($w \cdot x_i - b = 1 + \xi_i$) for the positive border and ($w \cdot x_i - b = -1 - \xi_i$) for the negative border.

$$
\mathcal{L}(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j K(x_i, x_j) \quad s.t. \ 0 \leq \alpha_{i,j} \leq C \quad (7.7)
$$

The learning in the primal form of Equation (7.5) is performed in the pure feature space which is formed by the vectors. Practically, the vector $w$ conveys information about which aspects of the feature space (vector entries) are more important for the candidate class. The dual form of the SVMs is recently used in which the kernel machine as presented previously performs the abstraction of the feature space into pairwise relationships in the kernel. The dual uses the labels $y_i, j$ and the kernel itself to maximise the Equation (7.7) by varying constraint Lagrangian multipliers ($\alpha_{i,j}$) [26]. The Lagrangian multipliers are the weights returned or the learned model by maximising the margin between the classes. These weights convey information about which points (→ vectors → images) are considered more or less important. The major difference with other classification models is that SVMs make use of the whole training set while disregarding any probabilistic priors as would a Bayesian model. In addition, convex constraint optimization approaches such as the Newton convergence and quadratic programming make the learning core of the SVMs one of the most resilient and robust classifiers [26], [23].

7.3.1 Experiments

Following a standard framework of image classification [58], spatial pyramids are equipped with the descriptors which were presented in Chapter
5. The benchmarked descriptors are the Gaussian foveal arrangement along with its sparse version and the LogFoveal arrangement with its sparse version. The performance of these descriptors is benchmarked on Caltech 101 and Pascal VOC 2011 databases. The training for both databases is performed using a linear kernel SVM. Specifically, for the Caltech 101, there are 10 training-testing cycles to establish the mean accuracy and standard deviation per class. Each class is trained on 30 random images selected from the same class (as positive examples) and 30 random images from the background-class (as negative examples). Once the SVM model per class is acquired then it is tested on 50 test images form the corresponding class that the model is trained and 50 randomly selected images from the background class. Next, the accuracy per model is monitored for 10 iterations. For the Pascal VOC 2011 database, 10 iterations per class assess the mean accuracy and standard deviation with 300 positive examples and 300 negative per class. The size of the codebooks generated for both databases is 500 visual words using $k$-means.

![Figure 7.3: This figure presents the classification accuracy of the Gauss-Foveal (D3 & D4) descriptors on Caltech 101.](image)

```python
Figure 7.3: This figure presents the classification accuracy of the Gauss-Foveal (D3 & D4) descriptors on Caltech 101.
```
Figure 7.3 graphically illustrates the classification accuracy of GaussFoveal (D3 & D4) descriptors and their sparse version over SIFT descriptors. The average classification performance is sorted in ascending order to clearly illustrate the differences in performance among the three descriptor types. The coloured areas indicate the standard deviation of the descriptors after 10 iterations of the classification process. Within the coloured regions, solid lines give the information for the average classification accuracy per class.

![Graphical representation of classification accuracy](image)

**Figure 7.4:** This is graphical representation of the classification performance of the LogFoveal (D5 & D6) descriptors on Caltech 101.

The next experiment shown in Figure 7.4, presents the classification accuracies of the LogFoveal (D5 & D6) descriptor arrangement and its sparse version on Caltech 101. The SIFT descriptors are used as a baseline performance to assess merits or disadvantages of the proposed descriptors. Both experiments show that these descriptor types perform better than SIFT.

Finally, the descriptors are tested on the Pascal VOC 2011 database with a similar training-testing protocol to the Caltech 101 database. The training phase involves kernels of 300 positive and 300 negative examples.
The negative examples are selected from all the classes except the training class (positive class). The testing is similarly performed for each class with one against all, as described the training phase. This process is repeated 10 times and the average accuracy and standard deviation per class are reported in Table 7.1.

<table>
<thead>
<tr>
<th>SP-SVM Accuracy</th>
<th>LogFoveal Sparse (D6)</th>
<th>LogFoveal Mag (D5)</th>
<th>SIFT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aeroplane</td>
<td>70.01±1.5</td>
<td>72.77±1.8</td>
<td>70.98±1.9</td>
</tr>
<tr>
<td>Bicycle</td>
<td>65.05±2.0</td>
<td>59.62±2.6</td>
<td>53.57±1.8</td>
</tr>
<tr>
<td>Bird</td>
<td>59.48±1.8</td>
<td>57.85±1.5</td>
<td>55.46±1.5</td>
</tr>
<tr>
<td>Boat</td>
<td>67.39±2.6</td>
<td>67.93±2.0</td>
<td>59.27±1.6</td>
</tr>
<tr>
<td>Bottle</td>
<td>55.86±1.4</td>
<td>55.41±1.6</td>
<td>54.60±2.2</td>
</tr>
<tr>
<td>Bus</td>
<td>75.18±1.6</td>
<td>67.52±1.9</td>
<td>61.58±2.3</td>
</tr>
<tr>
<td>Car</td>
<td>60.90±1.9</td>
<td>65.88±2.1</td>
<td>60.83±1.8</td>
</tr>
<tr>
<td>Cat</td>
<td>59.90±2.3</td>
<td>64.61±3.0</td>
<td>57.06±2.1</td>
</tr>
<tr>
<td>Chair</td>
<td>60.18±1.4</td>
<td>56.80±2.4</td>
<td>56.03±1.2</td>
</tr>
<tr>
<td>Cow</td>
<td>65.56±2.6</td>
<td>65.21±2.6</td>
<td>49.90±0.4</td>
</tr>
<tr>
<td>D-Table</td>
<td>63.10±1.2</td>
<td>54.67±2.0</td>
<td>52.45±0.8</td>
</tr>
<tr>
<td>Dog</td>
<td>60.43±2.2</td>
<td>63.22±1.9</td>
<td>57.81±1.9</td>
</tr>
<tr>
<td>Horse</td>
<td>55.80±1.0</td>
<td>56.55±1.7</td>
<td>54.61±1.5</td>
</tr>
<tr>
<td>Motorbike</td>
<td>65.36±1.6</td>
<td>62.39±2.0</td>
<td>54.25±1.2</td>
</tr>
<tr>
<td>Person</td>
<td>53.67±2.2</td>
<td>51.15±1.8</td>
<td>53.91±1.9</td>
</tr>
<tr>
<td>P-Plant</td>
<td>57.75±3.4</td>
<td>54.64±1.6</td>
<td>50.27±1.5</td>
</tr>
<tr>
<td>Sheep</td>
<td>68.60±2.3</td>
<td>68.59±2.8</td>
<td>58.53±1.8</td>
</tr>
<tr>
<td>Sofa</td>
<td>56.50±1.5</td>
<td>61.95±1.9</td>
<td>55.60±1.7</td>
</tr>
<tr>
<td>Train</td>
<td>65.88±2.0</td>
<td>59.69±1.5</td>
<td>52.01±1.7</td>
</tr>
<tr>
<td>Tv</td>
<td>64.96±2.4</td>
<td>63.21±2.7</td>
<td>58.00±1.7</td>
</tr>
</tbody>
</table>

Table 7.1: Spatial Pyramids and SVM Accuracy% in Pascal VOC 2011.

<table>
<thead>
<tr>
<th>Overall</th>
<th>AUC Caltech101</th>
<th>AUC Pascal VOC 2011</th>
<th>Accuracy Caltech101</th>
<th>Accuracy Pascal VOC 2011</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sparse LogGaussian (D6)</td>
<td>0.5630</td>
<td>0.5981</td>
<td>64.8±2.0</td>
<td>62.60±1.9</td>
</tr>
<tr>
<td>Sparse Gaussian (D7)</td>
<td>0.5413</td>
<td>0.5624</td>
<td>61.4±1.8</td>
<td>61.92±1.82</td>
</tr>
<tr>
<td>LogFoveal (D6)</td>
<td>0.5413</td>
<td>0.5449</td>
<td>64.00±1.7</td>
<td>61.48±2.07</td>
</tr>
<tr>
<td>Gaussian (D3)</td>
<td>0.5339</td>
<td>0.5428</td>
<td>64.02±1.99</td>
<td>60.21±1.92</td>
</tr>
<tr>
<td>SIFT</td>
<td>0.5258</td>
<td>0.5314</td>
<td>54.71±1.28</td>
<td>50.34±1.83</td>
</tr>
</tbody>
</table>

Table 7.2: Overall accuracies (averaged) across all 4 tests.

A summary of descriptors performance for different classification tasks is presented in Table 7.2. A first conclusion is that polar (or precisely foveated) arrangements perform consistently better than grid-based descriptors, such as SIFT. This is consistent with Chapter 5, where unoptimized bands of complex filters exhibit similar performance to Gaussian derivatives (SIFT). An important observation is that the scheme of enforcing sparsity does not improve performance in both databases. This might be due to different scale distributions in these databases. An observation that support this explanation is that the image size in Caltech 101
(approx. 250x300) is smaller than the Pascal VOC (approx. 400x500). High scale estimates are much less in number than the Pascal VOC scale estimates. Practically, a smaller image is saturated by blurring at much lower scales than the Pascal VOC images. This is an indication that enforcing the proposed sparsity scheme in Chapter 5 might not work for the fine scales of the filter bank, but only for the coarse ones. Alternatively, Pascal VOC images allow the keypoint detectors to produce more high scale estimates (coarse scales of the filter bank), where probably the sparsity scheme is useful.

### 7.4 Pairing Up Visual Words

Many modern approaches to vision problems, such as object recognition and categorization, are based on a series of well-defined processing stages which produce image representations that support either categorization or recognition. Typically, such implementations involve keypoint detection or dense sampling [36]. Commonly used keypoints detectors are the Harris and Hessian multiscale [68], MSER [66] and DoG [63]. These reduce the dense grid to a sparse representation, based around the keypoint locations and scales, yielding highly scalable performance. Recent research has shown that dense sampling can drastically increase the recognition performance through the inclusion of sparse coding and pooling strategies [96, 13]. For example, one selection mechanism for descriptors can combine sparse coding and max-pooling of features over image regions.

There are numerous methods that can also bind those mid-level descriptor based features into a compact representation while encoding spatial information. Methods for doing this include well-known spatial pyramids [58]. Learning networks employing pairwise potentials between edge features [59] and other higher-order spatial features [105] can also be employed. Spatial pyramids provide a description of relatively coarse spatial relationships, producing good recognition rates [58],[96],[13]. Geometric relationships can also be encoded by approaches which hypothesize a fixed number of features (parts). Part-based models have shown that fixed numbers of descriptor-based parts can be a powerful representation tool for object detection applications (see, for example, Felzenszwalb et al.[38]). These models have been tested in a variety of different arrange-
ments by [25]. Typical disadvantages of such models include having a fixed numbers of parts where a computationally expensive search assigns parts to locations and sets a hypothesised center for the model. Yet another approach binds descriptors into high-order features yielding much improved recognition performance compared to an ungrouped bag of features approach [106], [105]. The latter approach attempts to encode spatial relationships using relative feature distances (Correspondence Transform). Co-occurrences of these high order features are mapped into an offset space [105], where occurrence counts are assigned to these features which satisfy predefined distance criteria. Lastly another probabilistic approach for action recognition implements a Latent Dirichlet Allocation (LDA) variant with non-uniformity constraints to derive correlated feature groups which enhance the recognition of the framework [15]. This chapter focus on whether a selection mechanism can reduce the dictionary size of those high order features. The transition from visual words to 2nd, 3rd, ..., kth order grouping increases the computational complexity by a factor of \(N^k\), where \(N\) is the size of the visual word dictionary.

In this section, the use of kernels to create small-sized dictionaries of paired words utilizing *categorical opponency* to select such pairs is first proposed. Secondly, word pairs in close relative proximity are turned into a scalable indexing scheme, and the effect of different coupling functions on classification performance is studied. Thirdly, a method whereby coupling kernels is proposed, suggesting decision functions that can detect pair occurrences online. It is important to point out that the kernel in this section are strictly separate from the linear kernels used for classification.

### 7.5 Coupling Visual Words

The coupling of visual words is partly inspired by evidence that binding low-level features can improve recognition rates and pose invariance. Additional evidence has been found by Leordeanu, *et al.* [59] who used contour-like representations and Conditional Random Fields (CRFs) as a conveyor of spatial information, yielding improved classification. In addition, findings by Zhang, *et al.* [105] indicate that 2nd order features produce high performance gains compared to simple bag-of-visual-word approaches. Higher orders than the second, especially for the case in which an SVM is used for final classification, show almost no further
improvement [105]. Thus, this work is focussed on 2nd order features, or visual word pairs. Note that this is different to concatenations of descriptor pairs, which would in principle need to be clustered in a higher (double) dimensional-space. There is no work in the literature on the exact size of pair dictionaries used by computer scientists, or of the effect on performance. This motivates exploration of small codebooks of pairs with discriminative power that is, at least in principle, equal to that of a full paired dictionary of order $N^2$. For instance, a codebook of 500 visual words can produce $500^2 = 250,000$ pair combinations, but some of these are repeated. Assuming symmetry in the pairs, this yields a paired combination of $N(N - 1)/2 = 124,750$ unique pairs, or including the case that pairs with the same word IDs are allowed, $N(N + 1)/2 = 125,250$ combinations can be generated. This is an inconveniently large size for a dictionary produced by clustering alone, which discourages implementation when the dictionary needs to be regenerated in on-line use. Thus, a key question is: to what extent can word pairing produce an effectively much greater dictionary size, and provide the performance gains usually associated with large vocabularies of single-word dictionaries [73]?

7.5.1 A Dictionary of Paired Visual Words

Grouping visual words of a small dictionary size can exponentially increase the effective size of a visual vocabulary. Even paired words can yield numerous combinations, depending on the size of the initial codebook. Among these numerous pairs, a subset can exist that makes the pairing task more efficient. In addition, it is assumed that specific pairs are very unique to each class and others are non-informative. Thus, a pair mining method is described to select those pairs from the training set given the ground truth of the object classes.

One can treat pairing as estimating the joint occurrence of certain words, $w_i \cap w_j$, and consider the probability that these words occur jointly $P(w_i \cap w_j)$. Bayes’ Theorem suggests that $P(w_i \cap w_j) = P(w_i) \cdot P(w_j | w_i)$. However, it is intuitive to use a kernelized form which incorporates a term similar to a prior on individual word occurrence:

$$K_h(w_i, w_j) = P(w_j | w_i) \cdot G_i$$

and where $G_i = -\log(P(w_i))$ provides a weighting similar to an inverse-
The kernelized expression (7.8) provides a statistical means to monitor specific visual words down-weighted by the \( G \) term as introduced by Sivic, et al. \cite{87}. \( P(w_j | w_i) \) refers to the probability that codebook member \( w_j \) has occurred, given that the word \( w_i \) has occurred in the image. It was found that by stopping the bins corresponding to \( w_i \) in estimating \( P(w_j | w_i) \) reduces bias in (efficiently) estimating the co-occurrence of other words in the image. Specifically, the conditional histogram of visual words is constructed by removing the candidate \( w_i \) word-bin from the histogram. This procedure favours the pairing of words with those other than itself. The conditional word estimation is performed per object, then the kernels are averaged over the same class and finally \( L_1 \) normalized, arranged according to object classes.

\[
K_{Op}(w_i, w_N) = \frac{\max_{c_t} K_h(w_i, w_N, c_t)}{\max_{c_t \neq c_t} [K_h(w_i, w_N, c'_t) \leq \max_{c_t} K_h(w_i, w_N, c_t)]}
\] (7.9)

The parameter \( c_t \) characterises an object label assigned to a normalized kernel. Presumably, if a histogram of words can provide a rough discrimination among categories, then a pair might exist enhancing this behaviour. Expression (7.9) is one approach to detect which pairs have high dominance over a specific class. Searching along the categories \( c_t \in \{1, \ldots, C\} \) the maximum of a kernel entry is divided by the second maximum, found in another category. This forms a ratio such that the higher the value of the RHS of Equation (7.9), the higher the opponency.
of this class against the others. This means that for a given combination of visual words, the detected pair tends to be unique to the examined class.

7.5.2 Decision Functions for Coupling

In the training and testing phase, histograms of pair occurrences are derived based on the pair codebook described in Section 7.5.1. Zhang, et al.,[105] used a correspondence transform to build histograms of co-occurrences over predefined regions. However, their approach did not deal with the size variation of objects. For example, if salient patches on one object were to occur with a size such that multiple patches fall into the allowed offset space, a different result would be obtained if the object were scaled such that the relative distance between its salient patches was increased, despite the scale-invariance of typical descriptors. An alternative approach was used by H. Ling, et al.,[61], in which cumulative proximity distributions were employed. Both approaches lack the inclusion of scale-relative distances. So, the distances between descriptor pairs can vary as both the pose and size of the object change. Visual words, especially at the descriptor level, typically only convey information from a local region. However, the rough scale estimate provided by many of the approaches to building scale-invariant descriptors can be interpreted as the radius of a patch descriptor, and this should scale with patch zooms. A hypothesis can be made that these keypoint-associated scale estimates may provide reasonable scale predictions as an object is scaled, especially if one considers more than one patch. In addition, the relative distances of visual words in space is trivially obtained. It would appear sensible that the coupling functions should be based on the relative scale coverage that two visual words have between them. For instance, two words might slightly overlap each other, suggesting that the gradient fields encoded by both also jointly encode an object contour. In fact, such quasi-contour representations have shown very good recognition rates as discussed in Section 7.5.1. This leads to defining coupling functions that include relative distance to the effective descriptor radii. This transforms the relative overlap in image space into a probability of two co-occurring words forming a proximal pair.

An abstracted (“distilled”) feature \( \phi_w^{(n)} = (x_n, y_n, \sigma_n, w_j) \) is defined
Figure 7.6: This figure shows two word co-occurrences before pairing. The size of the circles indicates the scale at those locations, and the colour represents a unique word ID. The pair detection is performed by flagging the word IDs of a single pair. At this point, several word co-occurrences might happen. The kernel on the right of the figure, as defined in Equation (7.10) arranges the co-occurrences and selects the maximum values along the largest dimension. The remainders are added as probabilistic votes into the histogram bin which indicates the examined pair.

by its \((x_n, y_n)\) image location estimated by a detector while \(n\) represents the instance of a single word \(w_j\) in the image. The parameter \(\sigma_n\) is the scale estimate for the keypoint, and may be thought of as representing the descriptor radius. The \(w_j\) entry represents the index of the assigned visual word. Next the \(r = \sigma_n + \sigma_m\) is the summation of radii of two different features \(\phi^{(n)}_{w_j}\) and \(\phi^{(m)}_{w_i}\). Finally, the parameter \(d = \sqrt{(x_n - x_m)^2 + (y_n - y_m)^2}\) represents the Euclidean distance between candidate features. Often, in on-line image pair assignment, several co-occurrences \(\phi^{(n)}_{w_j} \cap \phi^{(m)}_{w_i}\) of the same words \(w_i\) and \(w_j\) may be found. A kernel of visual word co-occurrences is constructed in which the first dimension corresponds to the number of times word \(w_j\) is found \((N)\) and the second dimension, the number of times word \(w_i\) is found \((M)\). These are used in the construction of the word pairing histogram, \(\mathcal{B}^{(p)}\), defined by:

\[
\mathcal{B}^{(p)} = \sum_{n,m=1}^{N,M} \min(N,M) \max_{n,m=1} K_{N \times M}^{(p)}(\phi^{(n)}_{w_j} = w_j, \phi^{(m)}_{w_i} = w_i) \tag{7.10}
\]

A maximization operation is applied along the larger dimension. This approach was found to yield the best suited soft-assigned pairs which, in turn, will be added as pair occurrences into histogram bins \(\mathcal{B}^{(p)}\). The kernel mapping \(K_{N \times M}^{(p)}\) is obtained by applying one of the following three pairwise decision functions independently; in Section (3.1) we compare...
the different coupling criteria to assess which one yields the best performance:

\[ S_1(P_p|\phi^{(n)}_{w_j} \cap \phi^{(m)}_{w_i}) = \frac{1}{1 + e^{-\alpha \cdot d_{r}}} \] (7.11)

**Figure 7.7:** (a) Illustrates a word pair comprised of two words \(w_A\) and \(w_B\). These words convey information, such as the effective descriptor radius (\(\sigma_A\) and \(\sigma_B\) ), their relative Euclidean distance (\(d_{AB}\)), which is used to derive the paired-words. (b) Illustrates the number of times that a single pair occurred (as described in Equation (7.10) before adding up the votes.

Equation (7.11) expresses a sigmoid likelihood of the pairs \(P_p\) given at least one \(\phi^{(n)}_{w_j} \cap \phi^{(m)}_{w_i}\) co-occurrence happened. One advantage of using this function is that its output ranges always in \((0, 1)\) which in turn reduces the dynamic range of the descriptor. The raised term in the exponent represents the relative overlap of the two words. Also, the \(\alpha\) parameter affects the slope of the sigmoid curves; the higher the \(\alpha\) value the further away co-occurrences are paired up.

\[ S_2(P_p|\phi^{(n)}_{w_j} \cap \phi^{(m)}_{w_i}) = \max\{0, \frac{d_{r}}{r}\} \] (7.12)

The max() operation in Equation (7.12), rejects negative outputs from the \(\frac{d_{r}}{r}\) term. Occasionally, closely adjacent words might produce negative values. One would wish to penalise such occurrences, based on the idea that two distinct words with large overlap providing little new information. This function has no normalised output, hence distance “\(d\)” adds up to descriptor bins.

\[ S_3(P_p|\phi^{(n)}_{w_j} \cap \phi^{(m)}_{w_i}) = \sqrt{(x_n - x_m)^2 + (y_n - y_m)^2} \] (7.13)

Equation (7.13) is the typical Euclidean distance between words. This Equation is applied to identify the nearest word co-occurrences in image space. The effect of using Euclidean distance is to pair words that are adjacent to each other. These relative distances are accumulated to the
corresponding pair (histogram bin). The votes can capture trends that might exist regarding the relative distances. It is assumed that an SVM classifier can learn a separating hyperplane that splits these classes based on any underlying structure.

7.6 Experiments

The evaluation of the proposed approach starts with the size of a pair dictionary. In these experiments to determine the effect of dictionary size, the SIFT descriptor from VLFEAT [93] was used with both the built in detector and grid-based sampling. 128-dimensional SIFT descriptors were used, and for the case of the dense sampling, the patch size was $16 \times 16$ with 8-pixel spacing. For the first experiment, the single-word codebook was set to 500. The number of codewords in the paired codebook was then varied in size between 500 to just below 4000. One of the most important effects is on the variance of the classification accuracy with partial training sets. To determine this, a subset of training images (600 per class), was randomly sampled from the full Pascal VOC 2011 training database and used to learn a classification model using a linear SVM classifier [35]. A sample of validation images were drawn (600 per class) and accuracy was tested. By repeating this process 10 times per class, an estimate of the standard deviation is obtained. For dictionary sizes below 4000, the standard deviation is large, becoming greater as the number of word pairs is decreased. The standard deviation rose to as much as 20%, and this could significantly affect real-world performance. The Pascal VOC 2011 set was used because it contains slighter greater pose changes across many of the classes. Details of the classifier training are as follows: half of the kernel entries were randomly selected positive examples of the trained class and the other half were randomly selected negative examples. The overall size of the kernel is $600 \times 600$ entries balanced into positive and negatives which helps to obtain realistic accuracy estimates. Having determined that paired codebook sizes of beyond 4000 lead to standard deviations of less than 2.5%, a paired size of 5000 was selected for further experiments.
7.6.1 Evaluation of Coupling Functions

The performance of Equations (7.11), (7.12) is evaluated on the Pascal VOC 2011 as well. In this experiment, Equation (7.10) was used along with the different coupling functions. The same experimental setup, described above, is applied to assess which pairing kernels ($S_1$, $S_2$ or $S_3$) provide better performance.

<table>
<thead>
<tr>
<th>Equation</th>
<th>Average Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Eq. (7.13) $S_3$</td>
<td>63.92%</td>
</tr>
<tr>
<td>Eq. (7.12) $S_2$</td>
<td>65.93%</td>
</tr>
<tr>
<td>Eq. (7.11) $S_1$</td>
<td>66.42%</td>
</tr>
</tbody>
</table>

Table 7.3: This table clearly shows that the best performance is obtained by $S_1$ ($\alpha = 1$). The average performance of the examined measures was tested on the Train set which contains approximately 5K images.

![Variation of sigmoid slope vs Classification Performance Gain](image)

Figure 7.8: This is the only experiment in which the $\alpha$ parameter of Equation (7.11) is varied. A red line sets the baseline performance of the sigmoid function where the blue curve shows which values of $\alpha$ increase or reduce the accuracy. This test has been performed on the class “Cow” of Pascal VOC 2011. There is a clear trend for the specific range of $\alpha = (8, 10.5)$ which can yield better performance for this class. The nature of this effect is class specific.

7.6.2 Pascal VOC 2011

In this test, the performance of the best performing coupling kernel is compared ($S_1$) in all categories of the Pascal VOC 2011 dataset against the performance of spatial pyramids (Table 7.4, both keypoint and grid-based. The size of the histogram descriptors was 10500, using three levels
(0,1, and 2) [58]. The best performing method is shown in each column in bold. In Table 7.5, the average performance and standard deviation is summarised of the runs across the whole database when either pairs or single words are used, both with keypoint and grid-based approaches.

It is clear that the pairs also lend themselves to grid-based approaches, resulting in the highest performance in these tests.

<table>
<thead>
<tr>
<th>Average Accuracy</th>
<th>Aeroplane</th>
<th>Bicycle</th>
<th>Bird</th>
<th>Boat</th>
<th>Bottle</th>
<th>Bus</th>
<th>Car</th>
<th>Cat</th>
<th>Chair</th>
</tr>
</thead>
<tbody>
<tr>
<td>Keypoint-Based Pyramids (10500)</td>
<td>79.98</td>
<td>63.48</td>
<td>56.46</td>
<td>59.27</td>
<td>64.80</td>
<td>61.38</td>
<td>60.83</td>
<td>57.06</td>
<td>60.43</td>
</tr>
<tr>
<td>Grid-Based Pyramids (10500)</td>
<td>79.34</td>
<td>64.95</td>
<td>64.58</td>
<td>60.91</td>
<td>72.98</td>
<td>69.72</td>
<td>69.52</td>
<td>70.14</td>
<td>63.84</td>
</tr>
<tr>
<td>Keypoint-Based Pairs (5000)</td>
<td>76.06</td>
<td>59.48</td>
<td>70.98</td>
<td>68.30</td>
<td>68.30</td>
<td>73.60</td>
<td>84.25</td>
<td>76.10</td>
<td>66.95</td>
</tr>
<tr>
<td>Grid-Based Pairs (5000)</td>
<td>80.80</td>
<td>70.74</td>
<td>70.74</td>
<td>74.46</td>
<td>74.46</td>
<td>74.46</td>
<td>74.46</td>
<td>70.74</td>
<td>66.94</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Average Accuracy</th>
<th>Dog</th>
<th>Horse</th>
<th>Motorbike</th>
<th>Person</th>
<th>Plant</th>
<th>Sheep</th>
<th>Sofa</th>
<th>Train</th>
<th>Tvmonitor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Keypoint-Based Pyramids (10500)</td>
<td>77.00</td>
<td>65.38</td>
<td>65.38</td>
<td>55.61</td>
<td>50.27</td>
<td>58.85</td>
<td>71.00</td>
<td>67.45</td>
<td>74.25</td>
</tr>
<tr>
<td>Grid-Based Pyramids (10500)</td>
<td>67.82</td>
<td>64.11</td>
<td>63.83</td>
<td>66.45</td>
<td>59.66</td>
<td>61.24</td>
<td>70.94</td>
<td>73.45</td>
<td>74.25</td>
</tr>
<tr>
<td>Keypoint-Based Pairs (5000)</td>
<td>70.70</td>
<td>64.59</td>
<td>65.30</td>
<td>65.83</td>
<td>65.26</td>
<td>65.97</td>
<td>65.97</td>
<td>63.12</td>
<td>67.41</td>
</tr>
<tr>
<td>Grid-Based Pairs (5000)</td>
<td>69.68</td>
<td>65.68</td>
<td>63.18</td>
<td>59.41</td>
<td>58.85</td>
<td>61.88</td>
<td>74.02</td>
<td>59.57</td>
<td>74.22</td>
</tr>
</tbody>
</table>

Table 7.4: Detailed performance evaluation on Pascal VOC 2011.

In these experiments, the performance in a larger categorization database is assessed using keypoints and grid-based sampling with either a pair approach or a pyramid approach. In this case, training on 30 examples and testing on 50 randomly selected samples is performed. The single-word dictionary size is again 500, with the paired-word dictionary being 5000. A linear SVM was used as the classifier. In the case that a category contains less than 50 samples, testing is performed on the remaining number of samples. Figure 7.9 shows the classification accuracy. The word pairing significantly enhances results, which are summarised in Table (7.6).

<table>
<thead>
<tr>
<th></th>
<th>Pyramids (10500)</th>
<th>Pairs (5000)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Keypoint-Based</td>
<td>56.34±1.625</td>
<td>66.42±2.07</td>
</tr>
<tr>
<td>Grid-Based</td>
<td>67.52±2.036</td>
<td>69.53±2.03</td>
</tr>
</tbody>
</table>

Table 7.5: Summary of table (7.5). These are the accuracies averaged over all classes, with standard deviation reflecting variability across all classes.

7.6.3 Caltech101

In these experiments, the performance in a larger categorization database is assessed using keypoints and grid-based sampling with either a pair approach or a pyramid approach. In this case, training on 30 examples and testing on 50 randomly selected samples is performed. The single-word dictionary size is again 500, with the paired-word dictionary being 5000. A linear SVM was used as the classifier. In the case that a category contains less than 50 samples, testing is performed on the remaining number of samples. Figure 7.9 shows the classification accuracy. The word pairing significantly enhances results, which are summarised in Table (7.6).
Figure 7.9: In (a) a comparison of spatial pyramids (blue) and pair (red) with descriptors produced by keypoint locations. (b) The same comparison is illustrated, but this time a grid sampling approach has been implemented. In both (a) and (b) the “confidence” represents the unit-standard deviation envelope estimated over 10 runs.

<table>
<thead>
<tr>
<th></th>
<th>Pyramids (10500)</th>
<th>Pairs (5000)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Keypoint-Based</td>
<td>52.30 ±1.28</td>
<td>71.00 ±2.00</td>
</tr>
<tr>
<td>Grid-Based</td>
<td>60.16 ±1.56</td>
<td>73.88 ±2.10</td>
</tr>
</tbody>
</table>

Table 7.6: Average accuracies per method for Caltech 101.

### 7.6.4 Pascal VOC 2007

A test on the Pascal 2007 database was also performed using a larger single codebook dictionary of 4000 words. This test shows better performance in the spatial pyramid, and a comparable test of word pairing performance. The pyramid representation consisted of 84000-element descriptors, whilst 65000 word pairs were used. This was chosen as being roughly equivalent in performance improvement to the 5000 paired number relative to a 500 single codeword dictionary, i.e. $65000 \approx 4000^x$, where $x = \log_{500}(5000)$.

<table>
<thead>
<tr>
<th></th>
<th>Aeroplane</th>
<th>Bicycle</th>
<th>Bird</th>
<th>Boat</th>
<th>Car</th>
<th>Cat</th>
<th>Chair</th>
<th>Cow</th>
<th>Grid-Based Pyramids (84000)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dog</td>
<td>78.59</td>
<td>66.23</td>
<td>64.48</td>
<td>74.79</td>
<td>81.48</td>
<td>74.33</td>
<td>74.08</td>
<td>62.35</td>
<td>60.94</td>
</tr>
<tr>
<td>Horse</td>
<td>75.59</td>
<td>66.23</td>
<td>64.48</td>
<td>74.79</td>
<td>81.48</td>
<td>74.33</td>
<td>74.08</td>
<td>62.35</td>
<td>60.94</td>
</tr>
<tr>
<td>Motorbike</td>
<td>73.00</td>
<td>66.23</td>
<td>64.48</td>
<td>74.79</td>
<td>81.48</td>
<td>74.33</td>
<td>74.08</td>
<td>62.35</td>
<td>60.94</td>
</tr>
<tr>
<td>Person</td>
<td>72.94</td>
<td>63.49</td>
<td>62.94</td>
<td>64.24</td>
<td>61.00</td>
<td>64.02</td>
<td>64.39</td>
<td>61.39</td>
<td>60.94</td>
</tr>
<tr>
<td>Tvmonitor</td>
<td>72.94</td>
<td>63.49</td>
<td>62.94</td>
<td>64.24</td>
<td>61.00</td>
<td>64.02</td>
<td>64.39</td>
<td>61.39</td>
<td>60.94</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Dog</th>
<th>Horse</th>
<th>Motorbike</th>
<th>Person</th>
<th>Tvmonitor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Grid-Based Pairs (65000)</td>
<td>62.94</td>
<td>63.49</td>
<td>62.94</td>
<td>64.24</td>
<td>61.00</td>
</tr>
</tbody>
</table>

Table 7.7: Detailed performance on Pascal VOC 2007. Overall, the pair approach achieves 66.84% against 62.21% for the spatial pyramids.
In this section, the performance of the LogFoveal descriptors is obtained on the Pascal VOC 2011. The experiment is similarly established as in section 7.6.2. A keypoint detection method is employed as described in Chapter 4. The product of angular bands (4.1) is used to detect keypoints. The keypoint locations are sampled using the Sparse LogFoveal (D6) descriptor given the scale of each keypoint.

<table>
<thead>
<tr>
<th>Average Accuracy</th>
<th>Aeroplane</th>
<th>Bicycle</th>
<th>Bird</th>
<th>Boat</th>
<th>Bus</th>
<th>Car</th>
<th>Cat</th>
<th>Chair</th>
<th>Cow</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIFT Pairs (5000)</td>
<td>77.06</td>
<td>65.38</td>
<td>70.06</td>
<td>68.30</td>
<td>59.06</td>
<td>61.23</td>
<td>66.16</td>
<td>68.06</td>
<td>69.01</td>
</tr>
<tr>
<td>Sparse LogFoveal (D6) Pairs (4000)</td>
<td>79.70</td>
<td>69.40</td>
<td>68.53</td>
<td>75.04</td>
<td>57.60</td>
<td>81.11</td>
<td>68.66</td>
<td>72.46</td>
<td>66.51</td>
</tr>
<tr>
<td>Average Accuracy</td>
<td>Dog</td>
<td>Horse</td>
<td>Motobike</td>
<td>Oven</td>
<td>P-Plant</td>
<td>Sheep</td>
<td>Sofa</td>
<td>Train</td>
<td>TV Monitor</td>
</tr>
<tr>
<td>------------------</td>
<td>----------------</td>
<td>-------</td>
<td>---------</td>
<td>------</td>
<td>-----------</td>
<td>-------</td>
<td>------</td>
<td>-------</td>
<td>-----------</td>
</tr>
<tr>
<td>SIFT Pairs (5000)</td>
<td>70.70</td>
<td>64.70</td>
<td>55.60</td>
<td>65.63</td>
<td>65.26</td>
<td>70.32</td>
<td>68.35</td>
<td>66.32</td>
<td>67.04</td>
</tr>
<tr>
<td>Sparse LogFoveal (D6) Pairs (4000)</td>
<td>70.31</td>
<td>62.41</td>
<td>64.16</td>
<td>75.57</td>
<td>60.44</td>
<td>62.15</td>
<td>72.60</td>
<td>61.59</td>
<td>71.36</td>
</tr>
</tbody>
</table>

Table 7.8: Detailed performance on Pascal VOC 2011. In test the Sparse Log Foveal descriptors (D6) use smaller number of pairs than SIFT.

Descriptor (D6) uses a smaller number of pairs (4000) than SIFT (5000) which are obtain by the categorical opponency approach. The average performance for the Sparse LogFoveal is 68.6% and for SIFT reaches 66.42%. Although the small difference in the average performance (approximately 2.2%) LogFoveal descriptor (D6) outperforms the SIFT with a much lower numbers of pairs.
Chapter 8

Conclusions and Future Work

An object recognition framework is comprised of many components operating in sequence. Each component either improves selectivity and accuracy or adds an amount of invariance dealing with the changes in object appearance. A multi-stage approach was assumed in this thesis, and roughly split into Chapters. Each of the main chapters then suggests improvements and discusses the findings. This Chapter summarises and suggests future work aligned to the findings of Chapters 3-7. The following list is a summary of the processing stages considered:

1. **Front end filtering** stage may be gradient fields or complex filter responses as this thesis explores.

2. **Keypoint detection** (and scale estimation). Alternatively one may use dense sampling.

3. **Descriptor sampling**. A fundamental ingredient of fixed length local path description.

4. **Clustering**. Visual words are quantised groups of descriptors into a mean vector.

5. **Spatial information encoding**. Refer to broader information within an image than a local patch.

6. **Learning - Classification**. A method to separate classes.

8.1 Multilevel Complex Filtering

If we are to consider the three main biologically inspired methods, briefly described in Chapter 3, it is clear that the method that has been suc-
cessful, and adopted by the computer vision community to the greatest extent is the SIFT descriptor. This is surprising, because SIFT is the least biologically plausible, based on the fact that it only uses simple multiscale gradient fields to analyse spatial structure. What are the main reasons behind the success of the SIFT descriptor for object recognition?

SIFT is considered to be very easy to use, and very efficient. There are two main reasons – each image is decomposed into a collection of fixed-length descriptors, which could represent patches of very different size. This may be thought of as a focus-of-attention mechanism, but it is applied in such a way as to produce very convenient descriptors. The second major reason is that the descriptors include a local scale and dominant orientation estimate.

Neither HMAX nor Gabor jet approaches have to date included a robust, compact descriptor at only key locations, preventing the highly scalable performance of SIFT. At the same time, it is thought that SIFT descriptors do not work well for face recognition, where Gabor jets excel [1]. This presents a problem to systems that are designed for analysing all images, irrespective of content: it suggests that we would have to use different front-ends depending on the image type.

If, instead, a keypoint-type approach could be added to Gabor-base representations, the same front-end (i.e. multiscale spatial filters) could be used for both face recognition and object recognition, provided that the performance is at least as good as SIFT and its variants.

8.1.1 Future Work

The proposed approach of front end filtering is efficiently implemented by a Fourier decomposition by which the proposed design and optimisation of the complex log-normal filters can be achieved. Although, careful design of filters throughout the experiments, these designs do not convey information of all the aspects of the Fourier space. Specifically, the filters can be extended by removing the constraint of constant bandwidths from already the proposed multi-scale approach. This will increase the redundancy of the generated filter responses leading to richer descriptions of the objects. Despite this, a multi-band approach would require more computational power leaving this approach to future work.
8.2 Keypoint Detection

A method of keypoint detection and scale estimation was suggested and evaluated based on complex filter outputs. The keypoint approach relies on the raw filter magnitudes of each scale, combining the oriented outputs in two alternative ways. The oriented energies of a single scale are summed to produce a scale space where local maxima in \((x, y, \sigma)\) occur. These local maxima are considered repeatable under various scales of the same image allowing the scale invariance to be achieved. A second method, to produce keypoints from such a scale space is also examined by using the product of the oriented magnitude outputs. The localization of these points is performed using a weighted mean of the magnitude outputs of the log-normal filters. This allows sub-pixel and sub-scale estimation, producing reliable scale estimates, as shown in Figure 4.6. Due to frame properties, there a slight divergence of the scale estimates near to the borders of the frame. This is corrected by applying consecutive Gaussian 2D functions to tune and refine the filter’s bank scale selectivity. Specifically, the scale error of the estimates is corrected by learning the bias of the filter outputs in which small amounts of random Gaussian noise are introduced.

The repeatability of the detectors was tested under various affine deformations. The keypoint evaluation experiments in the Oxford Affine Detector test data show that the product of orientated magnitudes (4.1) yields the highest repeatability scores, with sum (4.2) being very close in performance. Another interesting property, revealed through these experiments, is that the product has enhanced orientation invariant behaviour comparing to the sum. In particular, the product (4.1) scale space yields higher scores when planar rotation is incorporated in the test. Finally, the repeatability scores are, on average, quite low for all methods, indicating that the scale and orientation invariance of the keypoints do not reach performance of 90% repeatability that even though maxima detection in scale space is a good approach, it might be improved. This leads to the suggestion of other methods capable of achieving higher repeatability scores with an example being the exploration of the underlying mechanisms of saccadic scan-paths and different focus-of-attention approaches.

Overall, this section considered how one can achieve keypoint detec-
tion derived from Scale Spaces built by log-normal complex filters. The results shown that repeatability scores outperformed typical keypoint detection approaches. This is quite favourable to complex filter approaches which would potentially replace the typical estimation of multiscale gradient fields and their keypoint approaches by more sophisticated methods yielding better performance than a Gaussian scale space.

8.2.1 Future Work

Even though the proposed keypoint detection and scale estimation outperform commonly used detectors, the overall performance is considered low. As mentioned previously, saccadic scan-paths may be considered an approach to improve image matching. One approach to this is to employ a graph approach to describe the scan-paths as spanning trees.

8.3 Descriptors

Local pooling strategies attempt to seek effective pooling spatial arrangements of gradient fields. Some of these strategies are tested in this section on complex filter outputs. The experiments addressed whether such responses perform better than a typical implementation of a Gaussian scale space with directional derivatives. The experiment in Table 5.1 showed that complex Gabor filters perform better than a typical SIFT implementation using common DoG detector and equivalent descriptor size variation. Although these filters were not optimized for classification performance, they marginally outperformed the SIFT descriptor.

The experiment in Figure 5.4 involves the optimisation of the LogFoveal arrangements. This experiment attempted to find the best spacing of the pooling regions which are deployed by the LogFoveal arrangements. It was found that this exhaustive optimisation exhibited class specific optima. Thus, the average performance from all classes was employed to yield a generalised arrangement of the LogFoveal arrangements. The optimised form yield marginal improvements in comparison with the proposed optimisation in this section.

The experiments, in Figure 5.7 and Tables 5.2 and 5.3 was performed by harvesting the magnitude outputs of complex log-normal responses with different amounts of band overlap. This was done by restricting
the radial and angular bandwidth of the filter to the indicated redundancy level in Table 5.2 for each test. These responses were harvested by rectangular SIFT descriptors at the exact scale spacing and comparable descriptor size per scale. The results showed that log-normal scale space outperformed the Gaussian approach with directional derivatives. Also, this test revealed that the level of redundancy (overlap of distributions) exhibits a class specific nature. Hence, average performance was employed to address the scale space performance. Next, the best three redundancy levels are selected to further test the effect of maximal suppression module.

Finally, foveal arrangements using either log-normal or Gaussian weighting functions were applied to the log-normal scale space. In these experiments (see Table 5.4 and Figure 5.8), yielding the descriptors LogFoveal (D5 & D6) and Gaussian (D3 & D4) LogFoveal descriptors performed better than Gaussian spatial weighting in many cases. Also, the maximal suppression again improved the descriptor performance over raw outputs. The same methodology was repeated on Caltech 101, where LogFoveal descriptors improve marginally over Gaussian. The improvement was consistent in both databases. In addition, the descriptor performance was further improved when the maximal suppression was incorporated into the descriptor construction. Overall, log-normal space behaved better than Gaussian scale space in the frequency and spatial domain becoming an attractive approach to descriptor sampling methodology as well.

The experiments suggest that redundancy affects the descriptor performance of the foveated pooling functions. Maximal suppression always yielded improvement over raw filter outputs for any spatial pooling technique and weighting function. LogFoveal descriptors showed better average performance than other approaches.

8.3.1 Future Work

Throughout these experiments, it was found that foveal pooling arrangements perform better than grid-based approaches. The choice of the pooling functions did not affect the performance dramatically. One may consider a completely different approach of pooling, such as a second stage filtering on the filter responses. Even though would need additional com-
putational power, the exploration of pooling gradients in descriptors is considered exhausted making one to explore a radical method of description.

8.4 Visual Words

Clustering approaches, and specifically vector quantization, attempts to find representative clusters or quantized vectors (visual words or cluster centres) among the original observations. The clustering approach used in this section is $k$-means with a fixed cluster number of 500 clusters. The choice of 500 visual words serves as a representative small size codebook for the previous graphical illustrations. Any larger codebook size would be inappropriate for illustration. Also, the small size enhances grouping properties, which in turn assists human interpretation. Large codebooks usually develop tolerance to small changes in the patch level, becoming difficult to set up an illustrations of visual models.

The first test addressed the intra-inter cluster distance distributions where it was found that equisized clusters did not always occur. After clustering, it is found that descriptors producing high class discrimination, form a non-uniform tessellation in feature’s space (see Chapter 5 Tables). For instance away from the dense cluster populated areas, the distribution of the clusters expectationally decreases. This is an important indication that other tessellation methods (typically clustering) using different grouping criteria, e.g. by using different distribution criteria over the feature space may improve the clustering performance. Based on the cluster observations visualised by the PCA approach, a log-polar tessellation could be adopted which considers variable partitioning of the feature space but thought must be given on the choice of the similarity measures as well. Another interesting property was that maximal suppression caused the descriptors to group themselves into elongated distributions (see Figure 6.11). Even though this behaviour is unusual in feature space, it produced high class separability.

Finally, a more interpretable test was established to inspect typical cluster centres and their most diverse cluster members. These were visually illustrated to highlight any profound differences. Comparing the two (Figures 6.12 and 6.13), the effect of the phase invariant nature of complex filtering could be seen at the visual word level. This property
is found among the members of all clusters such as bars, curvature, T-junctions, etc. Also, the clusters exhibit a pseudo structural invariance in both scale spaces. This is more profound for the Complex log-normal scale space, where T-junctions may have occluded parts. On the contrary, clusters, of the SIFT descriptors do seem to learn illumination invariance at this processing level.

The properties of the clusters which lead to improved categorization performance were studied. $K$-means is the only algorithm which was implemented to perform clustering on the descriptors. The first experiment retrieves the distances within and among the clusters as an attempt to investigate the clusters’ variation across the feature space. The results of this experiment were followed by PCA, as a dimensionality reduction, in order to visualise the visual words in their feature space. This assisted the visual validation of specific hypothesis about the distance distributions. Finally, the derived clusters were mapped to the corresponding image patches to visually inspect the cluster members. Throughout these experiments, the descriptors that enhanced the class separability were found to be distributed in a non-uniform fashion. These findings lead to speculation on whether other clustering approaches might improve the overall performance of categorisation tasks.

### 8.4.1 Future Work

The exploration of the cluster properties showed the difference of what is learned between gradient fields and complex filter responses. Although clustering used as vector quantization, the findings showed that an unsupervised learning process occurs. This raises the question whether a more efficient unsupervised cluster approach can “polish” the cluster properties towards improvement. One may think a simplex space where a Dirichlet process can be applied under optimisation criteria which maintain that learning towards the exhibited results.

### 8.5 Spatial Information

Finally, word pairing was employed to a) evaluate the effect of different sizes of word-pairing dictionary well below that theoretically provided by the approximate upper number of word-pair combinations, where the subset of pairs is chosen by category opponency b) to investigate different
pairing functions and c) to construct a pair histogram representation efficiently, as the visual words and locations are produced for a query image. In addition to proposing and describing the approach, the performance of classification in a number of standard datasets, including Pascal VOC 2007, Pascal VOC 2011 and the Caltech 101 database was amended.

Categorical opponency was proposed to reduce the number of word-pairs from a theoretical vocabulary of at most \((N)(N + 1)/2\) to a size that is tunable by the user for image categorisation. This allows storage space to be reduced in representing histograms of word pairs. Category opponency makes use of labelled categories to select word pairs that are more suitable for category discrimination. The results show that the number of words can be flexibly chosen, but very small word-pair combinations can lead to high variance (=low reproducibility) in learning performance.

The coupling functions assign a weight to each paired histogram bin based on proximity of two paired words and the scale of the words. Three types of pairing functions, \(S_1\), \(S_2\) and \(S_3\) were proposed and evaluated. The results were best using \(S_1\), a sigmoidal function with one parameter which was fixed for all but one experiment.

The pairs have the ability to improve keypoint performance to the point that it is close to that of grid-based methods. This is an important saving: rather a than dense sampling of descriptors, or very large vocabulary sizes (through expensive clustering with a large number of cluster centres), word pairs allow a computationally efficient and flexible way of using the same single-word codebook to effectively achieve much higher vocabulary sizes through word pairing. The fact that combinations of word pairs can be selected to boost categorization performance helps keep the vocabulary size low, yet with good performance.

### 8.5.1 Future Work

In future work, it is intended to merge pyramids with word pairs to investigate whether there are further improvements in performance. In addition, recent approaches to sparse coding with max pooling [13] have shown major performance gains with respect to hard-word assignment. This encourages an exploration of sparse coding on structured dictionaries such as in our case where two words form a pair: there is a hidden
structure that it have not utilized yet of hierarchical information derived from our approach. For example, if two words form a pair then this can be treated as a rough tree structure comprised of a root node (Pair) and two children (two words).
Bibliography


133


